Project Acronym:	GRASP
Project Type:	IP
Project Title:	Emergence of Cognitive Grasping through Introspection, Emulation and Surprise
Contract Number:	215821
Starting Date:	01-03-2008
Ending Date:	28-02-2012

Deliverable Number:	D18
Deliverable Title :	Acquisition of contextual knowledge through observation of human hands in action
Type:	PU
Authors	A. Argyros, T. Asfour, H. Deubel, R. Dillmann, M. Do, D. Jonikaitis, P. Koutlemanis, N. Kyriazis, I. Oikonomidis, K. Papoutsakis, T. Sarmis, T. Schubert, K. Tzevanidis, X. Zabulis
Contributing Partners	FORTH, LMU, KIT

Contractual Date of Delivery to the EC:	28-02-2011
Actual Date of Delivery to the EC:	28-02-2011

Contents

Chapter 1

Executive summary

Deliverable D18 presents the third year developments within WP1 - "Learning to Observe Human Grasping and Consequences of Grasping". According to the Technical Annex, deliverable D11 presents the activities in the context of Tasks 1.3 and 1.4:

- [**Task 1.3**] Observing humans: Definition and development of a system that detects and tracks humans and their movements in particular. Activities in this task will focus on the important problem of acquiring real 3D motion of the arms while the human is interacting with objects. The tracking should be successful also in cases when the robot does not have a frontal view of the human.
- [**Task 1.4**] Observing human grasping: Definition and development of a computational method that detects, tracks and represents human hands in action. The derived representation includes aspects and features in the full 4D spatiotemporal space (3D space and time dimensions). The aim is to extract from a sequence of stereoscopic hand observations, the information that is necessary and sufficient for subsequent (WP2) parsing and interpretation of observed hand activities that, in turn, support future repeats by a robotic hand. Activities within this task will address important subproblems such as figure-ground segmentation (environmental modelling, motion/colour based segmentation, coarse object categorisation) tracking humans/hands in 2D/3D (feature selection, hand models, representation of prior knowledge of motion models, prediction and search strategies), etc.

The work in this deliverable relates to the following third year Milestone:

- [**Milestone 7**] Observing consequences of grasping; vocabulary of robot action/interactions and definition of a hierarchical structure of features.

Still, the WP1 work carried out during the 3rd year is highly relevant to other project milestones:

- [**Milestone 2**] Definition of initial ontology based on human studies; acquisition (perception and formalisation) of knowledge through hand-environment interaction.
- [**Milestone 4**] Analysis of action-specific visuo-spatial processing, vocabulary of human actions/interactions for perception of task relations and affordances.
- [**Milestone 6**] Integration and evaluation of human hand and body tracking on active robot heads, demonstration of a grasping cycle on the experimental platforms.

The progress in WP1 is presented in the below summarized scientific publications, attached to this deliverable.

- In Attachment A, it is shown that dual-task costs are observed when people perform two tasks at the same time. It has been suggested that these costs arise from limitations of movement goal selection when multiple goal-directed movements are made simultaneously. To investigate this, we asked participants to reach and look at different locations while we varied the time between the cues

to start the eye and the hand movement between 150 ms and 900 ms. In Experiment 1, participants executed the reach first, and the saccade second, in Experiment 2 the order of the movements was reversed. We observed dual-task costs - participants were slower to start the eye or hand movement if they were planning another movement at that time. In Experiment 3, we investigated whether these dual-task costs were due to limited attentional resources needed to select saccade and reach goal locations. We found that the discrimination of a probe improved at both saccade and reach locations, indicating that attention shifted to both movement goals. Importantly, while we again observed the expected dual task costs as reflected in movement latencies, there was no apparent delay of the associated attention shifts. Our results rule out attentional goal selection as the causal factor leading to the dual-task costs occurring in eyehand movements.

- In attachment B, When reaching for objects, people frequently look where they reach. This raises the question of whether the targets for the eye and hand in concurrent eye and hand movements are selected by a unitary attentional system or by independent mechanisms. We used the deployment of visual attention as an index of the selection of movement targets and asked observers to reach and look to either the same location or separate locations. Results show that during the preparation of coordinated movements, attention is allocated in parallel to the targets of a saccade and a reaching movement. Attentional allocations for the two movements interact synergistically when both are directed to a common goal. Delaying the eye movement delays the attentional shift to the saccade target while leaving attentional deployment to the reach target unaffected. Our findings demonstrate that attentional resources are allocated independently to the targets of eye and hand movements and suggest that the goals for these effectors are selected by separate attentional mechanisms.

- In Attachment C, we investigated the effects of visuospatial attention on movement kinematics by employing a dualtask paradigm. Participants had to grasp cylindrical objects of different sizes (motor task) while simultaneously identifying a target digit presented at a different spatial location within a rapid serial visual presentation (perceptual task). The grasping kinematics in this dualtask situation were compared with those measured in a single task condition. Likewise, the identification performance was also measured in a singletask condition. Additionally, we kept the visual input constant across conditions by asking participants to fixate. Without instructions about the priority of tasks (Experiment 1) participants showed a considerable drop of identification performance (perceptual task) in the dualtask condition. Regarding grasping kinematics, the concurrent perceptual task resulted in a less accurate adaptation of the grip to object size in the early phase of the movement, while movement times and maximum grip aperture were unaffected. When participants were instructed to focus on the perceptual task (Experiment 2), the identification performance stayed at about the same level in the dualtask and the singletask conditions. The perceptual improvement was however associated with a further decrease in the accuracy of the early grip adjustment. We conclude that visual attention is needed for the effective control of the grasp kinematics, especially for a precise adjustment of the hand to object size when approaching the object.

- In Attachment D, we present a novel method that, given a sequence of synchronized views of a human hand, recovers its 3D position, orientation and full articulation parameters. The adopted hand model is based on properly selected and assembled 3D geometric primitives. Hypothesized configurations/poses of the hand model are projected to different camera views and image features such as edge maps and hand silhouettes are computed. An objective function is then used to quantify the discrepancy between the predicted and the actual, observed features. The recovery of the 3D hand pose amounts to estimating the parameters that minimize this objective function which is performed using Particle Swarm Optimization. All the basic components of the method (feature extraction, objective function evaluation, optimization process) are inherently parallel. Thus, a GPU-based implementation achieves a speedup of two orders of magnitude over the case of CPU processing. Extensive experimental results demonstrate qualitatively and quantitatively that accurate 3D pose recovery of a hand can be achieved robustly at a rate that greatly outperforms the current state of the art.

- In Attachment E, we start by observing that due to occlusions, the estimation of the full pose of a human hand interacting with an object is much more challenging than pose recovery of a hand observed in isolation. In this work we formulate an optimization problem whose solution is the 26-DOF hand pose together with the model parameters and pose of the manipulated object, that jointly best explain the incompleteness of hand observations resulting from occlusions due to hand-object interaction. Thus, occlusions is not a curse we bypass but a feature we exploit. The

proposed method is the first that provides accurate and fast solution to this problem. Additionally, it is the first to demonstrate that hand-object interaction is not necessarily a complicating factor but a context that can be exploited effectively for hand pose estimation. Extensive quantitative and qualitative experiments with simulated and real world image sequences as well as a comparative evaluation with a state-of-the-art method for pose estimation of isolated hands, support the above findings.

- In Attachment F, we present a fingertip tracking framework which allows observation of finger movements in task space. By applying a multi-scale edge extraction technique, an edge map is generated in which low contrast edges are preserved while noise is suppressed. Based on circular image features, determined from the map using Hough transform, the fingertips are accurately tracked by combining a particle filter and a subsequent mean-shift procedure. To increase the robustness of the proposed method, dynamical motion models are trained for the prediction of the finger displacements. Experiments were conducted on various image sequences from which statements on the performance of the framework can be derived.

- In Attachment G, we propose a novel, fully automatic method for the tuning of foreground detection parameters in calibrated multicamera systems. The proposed method requires neither user intervention nor ground truth data. Given a set of such parameters, we define a fitness function based on the consensus built from the multicamera setup regarding whether points belong to the scene foreground or background. The maximization of this fitness function through Particle Swarm Optimization leads to the adjustment of the foreground detection parameters. Extensive experimental results confirm the effectiveness of the adopted approach.

- In Attachment H, we present work on exploiting modern graphics hardware towards the real-time production of a textured 3D mesh representation of a scene observed by a multicamera system. The employed computational infrastructure consists of a network of four PC workstations each of which is connected to a pair of cameras. One of the PCs is equipped with a GPU that is used for parallel computations. The result of the processing is a list of texture mapped triangles representing the reconstructed surfaces. In contrast to previous works, the entire processing pipeline (foreground segmentation, 3D reconstruction, 3D mesh computation, 3D mesh smoothing and texture mapping) has been implemented on the GPU. Experimental results demonstrate that an accurate, high resolution, texture-mapped 3D reconstruction of a scene observed by eight cameras is achievable in real time.

- In Attachment I, we introduce a new method for integrated tracking and segmentation of a single non-rigid object in an monocular video, captured by a possibly moving camera. A closed-loop interaction between EM-like color-histogram-based tracking and Random Walker-based image segmentation is proposed, which results in reduced tracking drifts and in fine object segmentation. More specifically, pixel-wise spatial and color image cues are fused using Bayesian inference to guide object segmentation. The spatial properties and the appearance of the segmented objects are exploited to initialize the tracking algorithm in the next step, closing the loop between tracking and segmentation. As confirmed by experimental results on a variety of image sequences, the proposed approach efficiently tracks and segments previously unseen objects of varying appearance and shape, under challenging environmental conditions.

Appendix A

Attached papers

[A] D. Jonikaitis, T. Schubert, H. Deubel, Preparing coordinated eye and hand movements: Dual-task costs are not attentional, Journal of Vision (2010) 10(14):23, 117.

[B] D. Jonikaitis, T. Schubert, H. Deubel, Independent Allocation of Attention to Eye and Hand Targets in Coordinated Eye-Hand Movements, Psychological Science, published online 26 January 2011.

[C] C. Hesse1, H. Deubel, Accurate grasping requires attentional resources, submitted to Vision Research, under revision.

[D] I. Oikonomidis, N. Kyriazis, A.A. Argyros, Markerless and Efficient 26-DOF Hand Pose Recovery, in Proceedings of the 10nth Asian Conference on Computer Vision, ACCV2010, Queenstown, New Zealand, Nov. 8-12, 2010.

[E] I. Oikonomidis, N. Kyriazis, A.A. Argyros, Full pose estimation of a hand interacting with objects: Exploiting context to turn occlusions into a useful visual cue", work in progress.

[F] M. Do, T. Asfour, R. Dillmann Particle Filter-Based Fingertip Tracking with Circular Hough Transform Features, Accepted to Machine Vision Applications Conference, MVA 2011, Nara Japan, June 2011.

[G] K. Tzevanidis, A.A. Argyros, Unsupervised learning of background modeling parameters in multi-camera systems, Computer Vision and Image Understanding 115 (2011) 105116.

[H] K. Tzevanidis, X. Zabulis, T. Sarmis, P. Koutlemanis, N. Kyriazis, A.A. Argyros, From multiple views to textured 3D meshes: a GPU-powered approach, in Proceedings of the Computer Vision on GPUs Workshop, CVGPU2010, In conjunction with ECCV2010, Heraklion, Crete, Greece, Sep. 10, 2010.

[I] K. Papoutsakis, A.A. Argyros, Object tracking and segmentation in a closed loop, in Proceedings of the International Symposium on Visual Computing, ISVC2010, Advances in Visual Computing, Lecture Notes in Computer Science, vol. 6453, pp. 405-416, Las Vegas, USA, Nov. 29-Dec. 1, 2010.

Preparing coordinated eye and hand movements: Dual-task costs are not attentional

Donatas Jonikaitis
Allgemeine und Experimentelle Psychologie, Ludwig-Maximilians-Universität, München, Germany

Torsten Schubert
Allgemeine und Experimentelle Psychologie, Ludwig-Maximilians-Universität, München, Germany

Heiner Deubel
Allgemeine und Experimentelle Psychologie, Ludwig-Maximilians-Universität, München, Germany

Dual-task costs are observed when people perform two tasks at the same time. It has been suggested that these costs arise from limitations of movement goal selection when multiple goal-directed movements are made simultaneously. To investigate this, we asked participants to reach and look at different locations while we varied the time between the cues to start the eye and the hand movement between 150 ms and 900 ms. In Experiment 1, participants executed the reach first, and the saccade second, in Experiment 2 the order of the movements was reversed. We observed dual-task costs—participants were slower to start the eye or hand movement if they were planning another movement at that time. In Experiment 3, we investigated whether these dual-task costs were due to limited attentional resources needed to select saccade and reach goal locations. We found that the discrimination of a probe improved at both saccade and reach locations, indicating that attention shifted to both movement goals. Importantly, while we again observed the expected dual-task costs as reflected in movement latencies, there was no apparent delay of the associated attention shifts. Our results rule out attentional goal selection as the causal factor leading to the dual-task costs occurring in eye–hand movements.

Keywords: attention, eye movements, spatial vision

Citation: Jonikaitis, D., Schubert, T., & Deubel, H. (2010). Preparing coordinated eye and hand movements: Dual-task costs are not attentional. *Journal of Vision, 10*(14):23, 1–17, http://www.journalofvision.org/content/10/14/23, doi:10.1167/10.14.23.

Introduction

In everyday situations, we frequently reach for objects—be it a simple task like picking up a cup of coffee or a complex task like clearing an office table. However since we usually look at the object we reach for, most reaching movements actually require doing two things at the same time, that is, planning and executing an eye and a hand movement simultaneously (Horstmann & Hoffmann, 2005; Johansson, Westling, Backstrom, & Flanagan, 2001; Land & Hayhoe, 2001; Pelz, Hayhoe, & Loeber, 2001). It might seem trivial to plan both eye and hand movements together, but it constitutes an instance of cognitive multitasking.

It is known that doing two tasks simultaneously bring costs, since both error rates and reaction times typically increase as compared to doing only one task at a time. These are typically referred to as dual-task costs (Pashler, 1994; Schubert, 2008), which arise when two different tasks compete for limited cognitive resources. In such a scenario, the limited resources could either be shared between the two tasks, leading to a slowing of both (Kahneman, 1973), or else execution of one of the tasks could be postponed until critical processing in the other is finished (Pashler, 1994; Schubert, 1999). While much is known about dual-task costs and the situations in which they arise, it remains debated whether these do occur in the case of simultaneous eye and hand movements, and if so, which particular processing stage(s) between early stimulus processing and final execution of the movement might be specifically involved in the processing bottleneck.

A number of studies have shown that whether there is interference between eye and hand movements depends on a variety of factors, such as on how the saccade is elicited and on what type of manual response is required. Pashler, Carrier, and Hoffman (1993) have demonstrated that there are almost no dual-task costs when simple button presses and eye movements to an abrupt onset are prepared together, suggesting that reflexive saccades directed toward an onset stimulus can be possibly executed without interference. Similarly, no dual-task

doi: 10.1167/10.14.23 Received July 13, 2010; published December 20, 2010 ISSN 1534-7362

costs have been reported when people made reflexive saccades to a peripheral location and simultaneously performed a rhythmic manual tapping task (Sharikadze, Cong, Staude, Deubel, & Wolf, 2008). In contrast, dual-task interference was found to occur even with simple button presses when non-reflexive saccades had to be performed to a location indicated by a central cue (Pashler et al., 1993); obviously, the planning of these saccades required an intentional selection of the movement goal.

Dual-task interference becomes more prominent when, instead of a simple button press, a manual reaching movement is required. It has been observed that latencies of saccades directed to peripheral onsets are longer if, simultaneous to saccade preparation, a reaching movement has to be planned to the same target (Bekkering, Adam, Kingma, Huson, & Whiting, 1994; Bekkering, Adam, van den Aarssen, Kingma, & Whiting, 1995). This suggests that dual-task costs for saccades arise when a reach must be directed to a spatial target, but not when the movement involves just a simple (non-spatial) button press. In other words, it seems that dual-task costs do arise when both eye and hand movements rely on the selection of a spatial movement goal. They also arise when saccades and button press responses have the same or a different directional component (e.g., to make a saccade to the right and press the left button; Huestegge & Koch, 2009). These findings make it likely that the mutual interference between the two tasks occurs in the movement planning phase, for instance, during the selection of the movement target (Bekkering et al., 1995), rather than in movement execution. Movement goal selection ("I will reach for this apple") occurs at an early stage of movement planning during which object parameters such as target location in space and object size are specified (Andersen & Buneo, 2002; Milner & Goodale, 1995).

While at least some of the dual-task costs can be explained by assuming that the two effectors compete for resources to select the movement goal, not all findings suit this pattern. Some studies reported even shorter saccade or reach latencies if participants made simultaneous eye and hand movement to the same object as compared to making single eye or hand movements (Lünenburger, Kutz, & Hoffmann, 2000; Niechwiej-Szwedo, McIlroy, Green, & Verrier, 2005). However, these observations do not necessarily contradict the hypothesis that movement goal selection leads to dual-task costs, since in all of the above-mentioned studies movement goal selection was limited by the fact that participants were asked to make eye, hand, or both movements to only one common target present on the screen (or to one of two targets present in separate visual hemifields). This raises the question as to which degree saccade or hand movement goal selection was activated, since in some cases movements might have been purely reflexive, toward a single target present within one visual hemifield.

It is important to note that none of these studies analyzed movement goal selection directly but instead relied on indirect measures such as reaction times or movement endpoint errors. Thus, it is possible that while movement execution was delayed in a dual-task situation, movement goal selection was not affected by the need to perform a second task.

It is now well established that the selection of a stimulus as the goal of a movement is related to attention shift to the movement target. A number of studies have shown that these attention shifts precede the initiation of goal-directed saccades, reaching movements and grasping (Baldauf & Deubel, 2010; Deubel & Schneider, 1996; Deubel, Schneider, & Paprotta, 1998; Montagnini & Castet, 2007; Schiegg, Deubel, & Schneider, 2003). Hence, spatial attention can be used as an index of movement goal selection before movement onset.

By measuring both movement latencies and spatial attention, we investigated whether movement goal selection is the causal factor that leads to the costs observed in these dual-task situations. Participants performed conjoint saccades and manual reaching movements while we manipulated the temporal overlap between the planning of these movements. In three experiments, two central movement cues were presented sequentially, with a variable stimulus-onset asynchrony (SOA) between the presentations. The movement cues could indicate either the same spatial location or spatially separate locations. The range of SOAs was selected such that in the short SOA conditions planning of saccade and reaches would overlap, whereas in the long SOA conditions those tasks would not overlap. If dual-task costs would occur, they should be largest at the shortest SOAs and smallest at the longest SOAs. In Experiment 1, the first cue specified the reach goal, and the second cue indicated the saccade goal. In Experiment 2, we measured whether dual-task costs are observed also when the movement order was reversed—the first movement cue indicated the saccade and the second cue specified the reach. Finally, in Experiment 3, we measured movement goal selection by using spatial attention as its index. Participants had to reach and look at two different locations while we presented a perceptual probe (a letter) at randomly chosen times during movement planning. This probe could be presented at cued saccade or reach locations, or at locations that were not relevant for the action. It is established that probe discrimination at exogenously or endogenously (as is the case with movement planning) cued locations can index attention at that location (for a review, see Carrasco, 2006). Thus, we could measure whether attention shifted to saccade or reach locations, and whether this shift was delayed when saccade and reach planning overlapped. Combined, the three experiments should reveal (1) whether there are dual-task costs for combined eye and hand movements as reflected in movement

latencies, and (2) whether these costs would arise due to movement goal selection as measured in probe discrimination at the saccade and reach goals.

Experiment 1

In Experiment 1, we determined the dual-task costs arising in a situation in which participants first made a reach, and then a saccade. We varied the time interval (SOA) between two arrow cues instructing to start each movement. If dual-task costs occur, the costs should be largest at the shortest SOAs and smallest at the longest SOAs (Schubert, 1999), since under the first conditions saccade and reach planning are temporally more overlapping than under the latter in which saccade planning starts long after the reaching onset. Additionally, we manipulated whether eye and hand movement goals were shared or not: on half of the trials, participants made saccades and reaches to the same location, and on the other half of trials to two different locations. If eye and hand movement planning shares a common goal selection process, then for the short SOAs there should be a crosstalk between these two systems, resulting in faster saccades and reaches when the two movements were directed to the same goal. On the other hand, if the goals for eye and hand movements are selected independently, there should be no benefit to plan saccades and reaches to the same location.

Methods

Participants

Twenty-two participants (mean age 23 years, 10 females) participated in this study. All participants had normal or corrected-to-normal vision. Informed consent was obtained from all participants.

Apparatus

Participants sat in a dimly illuminated room. They placed their right hand on a slightly inclined pointing plane, under a one-way mirror. Stimuli for pointing and saccades were projected from a monitor above onto the mirror. This setup allowed the visual stimuli to appear on the pointing plane, while the participants could not see their hand. In order to provide visual feedback about the hand position, an LED fixed to the tip of the right index finger could be illuminated during the experiment. LED was lit up in the beginning of the trial for participants to arrange their finger with visual stimulus and was illuminated at the end of the trials to provide feedback about reaching accuracy. Stimuli were presented on a 21-inch Conrac 7550 C21 display with frame frequency of 100 Hz, at a display resolution of 1024 * 768 pixels. Visual stimuli were shown on a gray background with a mean luminance of 5.1 cd/m^2.

Reaching movements were recorded with a Fastrack electromagnetic position and orientation measuring system (Polhemus, 1993), consisting of a central transmitter unit and a small receiver, mounted on the tip of the index finger of the participant's right hand. The sender unit was placed 60 cm in front of the participant. The device allows for a maximal translation range of 10 ft, with an accuracy of 0.03 in RMS. The frequency bandwidth of the system is 120 Hz; the time delay is 4 ms. Eye movements were recorded with a video-based eye tracking system (Senso-Motoric Instruments, Eyelink-I), which provides an accuracy better than 0.1 degree, at a recording frequency of 240 Hz. Head movements were minimized by means of an adjustable chin rest.

Procedure and stimuli

Figure 1 depicts the stimulus sequence. During each trial, a central fixation cross and twelve mask elements (size 0.9 × 1.4 deg, composed of randomly generated lines) were presented on the uniform background, arranged on an imaginary circle with a radius of 6.5 deg. Participants first directed the index finger of the right hand and their gaze to the central cross; 580 to 880 ms later, the first movement cue—an arrow that pointed toward one of the mask stimuli—was presented at the central fixation. The mask elements were arranged on the circle as if forming a clock face, and the arrow could point toward 2, 4, 8, or 10 o'clock. The arrow was presented for 100 ms, and participants were instructed to reach with the right index finger to the object indicated by this cue. After a variable time (SOA) from the first cue onset, a second movement cue was presented, again for 100 ms. Participants were instructed to saccade to the location indicated by the second arrow. On 50% of the trials, the second cue indicated the same target as the first cue (thus participants had to reach and look at the same location); on the other 50%, the second movement cue indicated a different target than the first cue. In those trials where the cues indicated different targets, the distance between the first and second movement targets was either three items in the clockwise direction or three items in the anti-clockwise direction (for example, if the first cue indicated a reach target at 2 o'clock, then the second cue would indicate (with equal probability) a saccade target at 5 or 11 o'clock, which amounts to an angular distance of 90 degrees from the first cued location). The SOA between the two movement cue onsets was 150, 200, 300, 350, 400, 450, 500, 600, 700, 800, or 900 ms. We chose this wide interval of SOAs in order to precisely measure at which cue delay dual-task costs would appear for saccades and reaches. Since the reaching latencies were typically 200–300 ms, the interval covered the time when reaches were still planned, when the hand

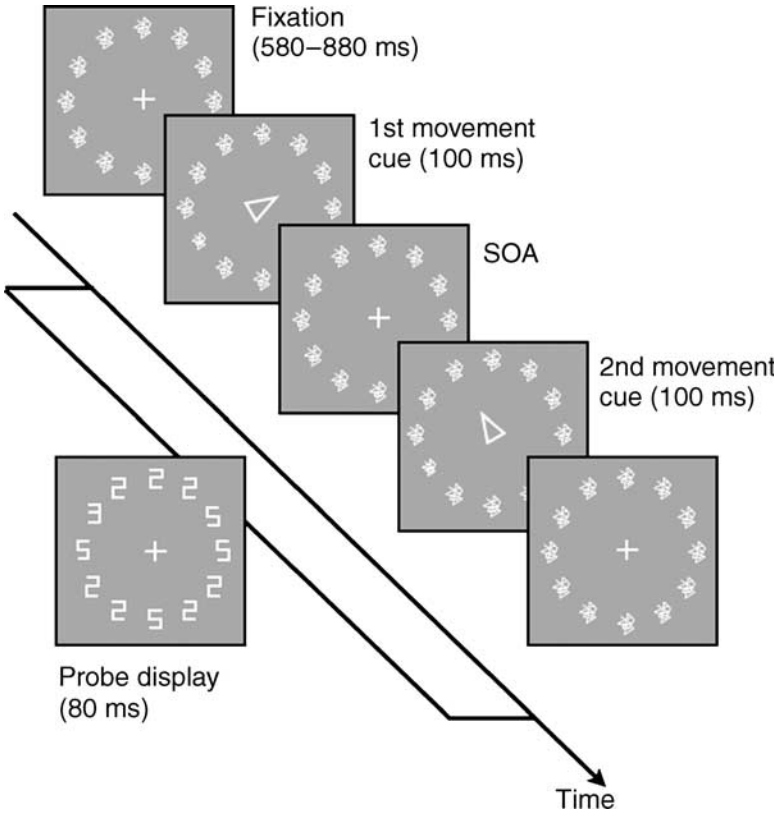


Figure 1. Experimental procedure. In Experiment 1, participants were asked to quickly reach to the location indicated by the first arrow and then quickly saccade to the location indicated by the second arrow. The time interval between the arrow onsets (SOA) was varied. In Experiment 2, the arrow appearing first instructed the saccade, while the second appearing arrow instructed the reaching. In Experiment 3, participants again reached and looked at two objects indicated by the two subsequent cues. Additionally, a probe display appeared at −50 to 650 ms with respect to the first movement cue onset. The probe was a digital letter "E" or "3," embedded in a circular array of distractors. Participants reported the probe identity after completing the movement task.

was in motion, and when the finger was already at the object.

Participants were instructed to reach and look as quickly as possible when the respective movement cues appeared, without delaying their movements or trying to group them. Each participant performed 4 experimental blocks of 144 trials each. All participants had a practice block before starting the experimental task. Six of the participants performed 12 experimental blocks in order to investigate possible practice effects.

Movement data analysis

Saccade and reach movement data were stored for offline analysis and saccades and reaches were detected using custom software. Reach onsets were defined as points in time when the vectorial velocity reached a threshold of 1°/s. Saccade onsets were defined as points in time when eye velocity threshold exceeded 150°/s. We further defined a 2° radius circle around central fixation as a maximum window within which saccade and reach movement starting position could vary. We removed all trials in which saccades smaller than 2° in size appeared before saccade cue onset. We accepted reach or saccade endpoint as correct if it fell closer to the reach/saccade goal than to any other irrelevant location besides the goal, and if the movement had a minimum latency of 100 ms after the movement cue appeared. Additionally, all trials with saccade or reach latencies longer or shorter than 3 standard deviations from the mean of each subject were rejected.

Results

We analyzed whether there were dual-task costs when participants made combined eye and hand movements. If there were no dual-task costs, then neither the reaction time of the first task (the reaching) nor the reaction times of the second task (the saccade) should be influenced by the SOA manipulation. Typical dual-task costs would be reflected in an effect of SOA on the reaction times of the second task (the saccade)—these should be longer for the short SOA conditions than for the long SOA conditions.

For the reaction time of the first task, there should be either no effect of SOA or an effect that should also depend on SOA.

The data indeed revealed that the SOA manipulation did not affect the reaction times of the first task—reaching latencies for the shortest SOA of 150 ms were 336 ± 14 ms (mean and standard error of the mean) and were 337 ± 15 ms for the longest SOA of 900 ms (repeated measures ANOVA, $F(10, 210) = 0.79$, $p > 0.6$). This means that participants started the reach movement immediately after the first movement cue appeared and did not try to postpone their response until the second movement cue was shown.

The SOA manipulation had a markedly different effect on the saccade latencies. Saccade latencies decreased with increasing SOA (repeated measures ANOVA, $F(10, 210) = 53.03$, $p < 0.01$), indicating that in the short SOA condition participants were not able to initiate their saccade immediately after the saccade cue appeared. The observed dual-task costs were about 100 ms—saccade latency decreased from 384 ± 14 ms for the 150-ms SOA condition to 280 ± 9 ms for the 900-ms SOA condition. Thus, typical dual-task costs did occur under these conditions, with participants being unable to perform the eye movement before they finished preparing the reaching movement.

We next analyzed whether there were any costs or benefits when saccades and reaches were directed to the same location or to different locations. First, we analyzed reaching movements, as it has been shown that in dual-task situation the task that is performed first (here, the reach) is completed faster if the second task shares a common response code (here, the saccade made to the same location as the reach), compared to a situation with different responses in the two tasks (Hommel, 1998; Lien & Proctor, 2002). Unexpectedly, we did not observe this effect—reaching latencies were not shorter when saccades and reaches were directed to the same location (Figure 2A; none of the planned one-tailed repeated measures *t*-tests comparing each time bin was significant, all *p*s > 0.5). This indicates that planning saccade and reach to the same location did not facilitate the preparation of the reach. One possibility of explaining this discrepancy is that we used a larger number of potential target locations (targets could appear at 8 different locations), unlike other studies (e.g., Hommel, 1998; Lien & Proctor, 2002) that used mostly two opposing response categories (e.g., left vs. right motor response). Furthermore, our task required precise spatial location coding—to reach to one of the multiple locations on the screen while making a saccade to a different location—instead of button presses. Note that the need to plan spatially directed movements and the number of potential reach locations could also interact, as reaches to displays with multiple objects are executed faster than reaches to displays with fewer objects (Song & Nakayama, 2006).

Next, we analyzed whether there were benefits when the saccade was directed to the same location as the reach. It

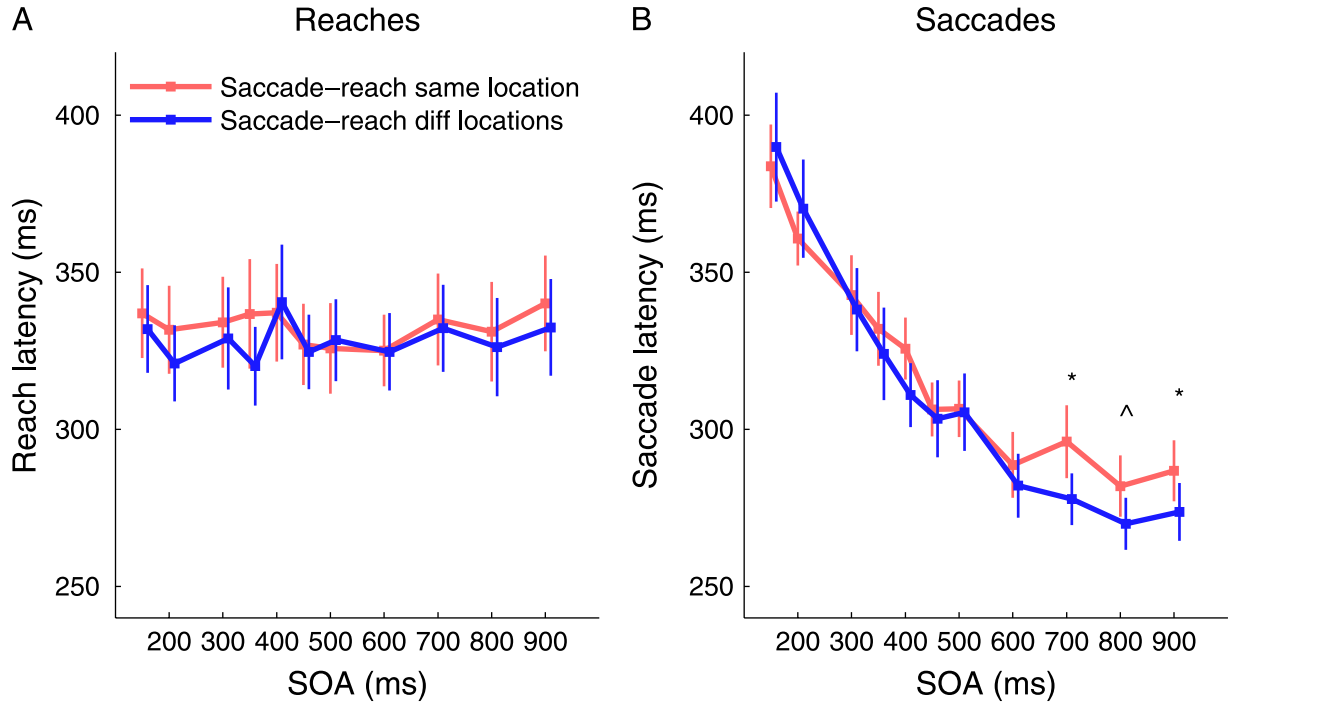


Figure 2. Dual-task interference in Experiment 1. (A) Latencies of the reaching movements as a function of SOA. (B) Saccade latencies as a function of SOA. Data are shown separately for trials when saccades and reaches were directed to the same location (red curves) or to different locations (blue curves). Symbols in (B): *$p < 0.05$, ^$p = 0.08$. Vertical bars indicate ±*SE*. Data are slightly translated horizontally to increase the visibility of different conditions.

has been demonstrated that movement planning leads to a shift of attention to the movement goal location (Deubel & Schneider, 1996; Linnell, Humphreys, McIntyre, Laitinen, & Wing, 2005); thus, planning a movement to one location is likely to be helpful as a cue in planning a subsequent movement to the same location. This leads to the prediction that latencies of saccades when they are directed to the same location as the reaching should be shorter than latencies of saccades directed to different locations than reaches. A two-way ANOVA with the first factor SOA and the second factor specifying saccade/reach location agreement did not show significant effect of the second factor ($F(10,210) = 1.51$; $p = 0.2$). However, interaction between the second and SOA and saccade/reach location agreement was significant ($F(10,210) = 2.42$; $p < 0.01$). We looked in more detail at short and long SOA conditions by performing separate *t*-tests. Our planned comparisons also showed that for SOAs less than or equal to 600 ms saccade latencies were not shorter if the saccades were directed to the same location as reaching, not even for the shortest 150-ms SOA condition (Figure 2B; at this time bin, mean latency of the saccades directed to the same direction as reaching was 384 ± 13 ms; mean latency of the saccades made to a different location than reaching was 390 ± 17 ms, repeated measures *t*-test $p = 0.60$).

For the long SOA conditions starting at 700 ms, saccade latencies were found to be even longer if saccades were directed to the same as compared to a different location than the reaching movement (Figure 2B, last three SOA conditions). A two-way ANOVA over these 3 last SOA conditions with second factor specifying saccade/reach location agreement was significant for the second factor ($F(1,21) = 13.21$, $p < 0.01$). This effect seemed to persist over all three SOA conditions (SOA 700, 800, and 900 ms; individual repeated measures *t*-tests), and individual subject data showed that majority of the subjects demonstrated this effect. This effect can possibly be attributed to Inhibition of Return (Klein, 2000), which we will discuss in more detail later.

Our findings demonstrate that the second (saccade) task was delayed while the first (reach) task was processed. In order to provide further evidence that reach planning indeed delayed saccades, we analyzed whether on trials with longer reach latencies the saccades also exhibited longer delays. For this purpose, reach latency in each trial was assigned to one of four quartiles (movement latencies increased from 260 ms in the first quartile to 380 ms in last quartile). Then, saccade latencies were separated into trials where the reaching latencies belonged to the 1st, 2nd, 3rd, and 4th quartiles. If reaching movements were delayed, then saccade latencies should be delayed as well. Thus, for short SOAs saccade latencies should be shorter if reaching latencies were shorter and longer if reaching latencies were longer. For long SOA conditions, this effect should disappear, as reaches would have already started or

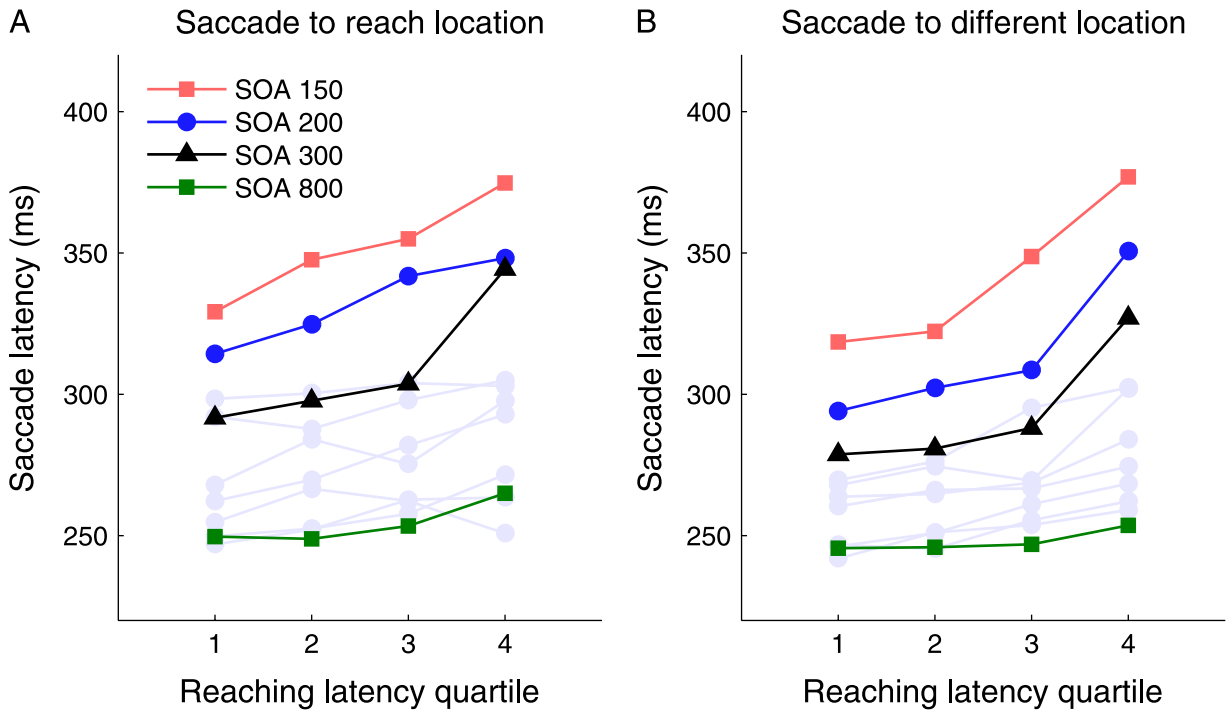


Figure 3. Reaching movements delay saccades. Data are shown for trials where the saccade was directed to (A) the same location as the reaching or to (B) a different location. Reaching latencies were divided into quartiles; the higher the quartile number, the longer the reaching latency. Four sample SOA conditions are shown (see figure legend). Other SOA conditions are plotted as light blue lines.

even finished. For this analysis, we again split the data according to reach/saccade location agreement. Figure 3 shows the result of this analysis, depicting saccade latencies for all SOA conditions. Figure 3A depicts the results for trials where reaches and saccades were directed to the same location, and Figure 3B depicts those for trials where the movements were aimed to different locations. The data show that longer reaching latencies indeed resulted in longer saccade latencies. This effect was most pronounced for the shortest SOA conditions. For SOA 150 ms, the saccades were about 50 ms slower for the longest as compared to the shortest reaching latency quartile, whereas in the SOA 800-ms condition this difference was only 20 ms.

We computed repeated measures two-way ANOVAs with quartile and SOA as main factors. We split this analysis for trials with saccades and reaches to the same location and trials with saccades and reaches to different locations. When reaches and saccades were directed to the same location, the main effect of SOA was significant, indicating that saccade latencies decreased with increasing SOA, $F(10, 150) = 7.93$, $p < 0.01$. The main effect of quartile was also significant, showing that longer reaching latencies lead to longer saccade latencies ($F(3, 150) = 18.12$, $p < 0.01$). The interaction between these two factors was also significant, $F(30, 150) = 1.61$, $p < 0.05$.

An equivalent analysis performed for trials when the saccade and reaches were directed to different locations revealed similar results. Again, saccade latencies decreased with increasing SOA (main effect of SOA was significant, $F(10, 150) = 8.88$, $p < 0.01$), and longer reaching latencies led to longer saccade latencies (main effect of quartile, $F(3, 150) = 27.65$, $p < 0.01$). The interaction between SOA and quartile was also significant ($F(30, 150) = 1.60$ $p < 0.05$), again meaning that longer reaching latencies delayed saccades most in the shortest SOA conditions.

We also analyzed whether longer reach latencies delayed saccades more or less, if saccades were directed to the same or different location as reaches. We found no significant differences between those conditions (paired samples *t*-test comparisons for saccades directed to the same versus saccades directed to different location than reaching for each reaching latency quartile were not significant, $p > 0.05$).

Last, we analyzed movement endpoint errors. When making saccades and reaches to two different locations, participants sometimes made movement errors by either looking at the reach goal (15% of trials in this condition) or by reaching to the saccade goal (14% of trials), implying a crosstalk between the movement planning for the hand and for the eye. We propose that these errors may result from the difficulty of our task in which two types of trials were interleaved—eye and hand movements directed to the same location or to different locations. Participants may have sometimes failed to switch to the less preferred type of task (eye and hand movements directed to different locations) and instead looked and reached to the same target.

Experiment 2

In the second experiment, we asked whether similar dual-task costs could be observed when the participants first made a saccade, and then a reach.

Methods

Seven participants (mean age 25 years, 3 females) participated in the study. All participants had normal or corrected-to-normal vision. Informed consent was obtained from all participants.

The procedure was the same as in Experiment 1, with the following exceptions. The first movement cue now directed the saccade, while the second movement cue directed the reaching movement. SOA between the cues varied between 150 and 600 ms (150, 200, 250, 300, 350, 400, 450, 500, and 600 ms). Each participant completed 3 blocks of 144 trials.

Results

After the first movement cue appeared, a saccade was initiated with a similar latency for all SOA conditions (repeated measures ANOVA, main effect of SOA not significant, $F(8, 48) = 0.37$, $p > 0.9$). Thus for the SOA 150-ms condition, i.e., the shortest SOA, mean saccade latencies were 314 ± 33 ms, which was not different from the longest SOA, the SOA 600-ms condition, in which saccade latencies were 334 ± 46 ms. In contrast, reaching latencies showed pronounced dual-task costs—as SOA increased, reaching latencies decreased ($F(8,48 = 8.05$, $p < 0.01$)). For the SOA 150-ms condition, mean reaching latency was 499 ± 27 ms, which was longer than for the SOA 600-ms condition in which mean reaching latency was 417 ± 25 ms ($t(6) = 6.06$, $p < 0.01$). Thus, in the present task the reach latencies revealed dual-task costs of around 80 ms (mean RT at SOA 150 ms − mean RT at SOA 600 ms). Figure 4 shows saccade and reach latencies as a function of SOA for trials when saccades and reaches were directed to the same location or to different locations. Again, saccade latencies were not shorter when saccades and reaches were directed to the same location (repeated measures *t*-test, all *p*s > 0.05). On the other hand, reach latencies were affected by saccade target

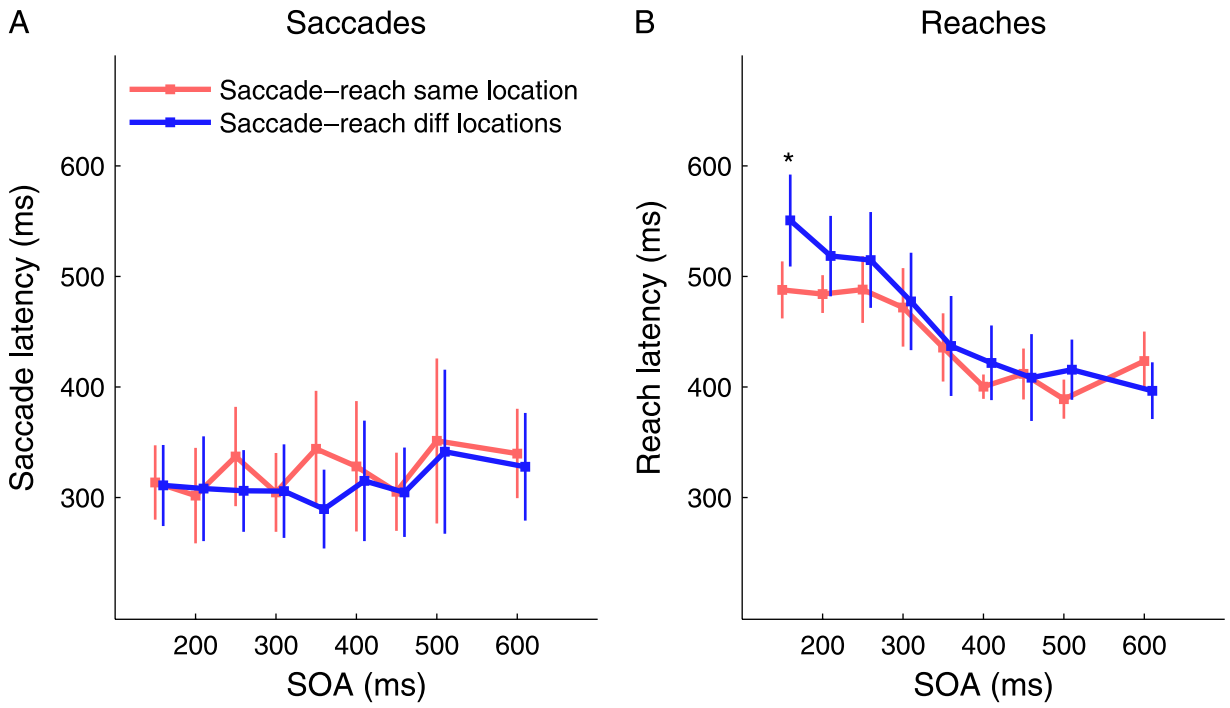


Figure 4. Dual-task interference in Experiment 2. The first movement cue directed the saccade target; the second movement cue instructed the reach target. Data are shown for trials when saccades and reaches were directed to the same location (red line) or to different locations (blue line). Symbol in (B): *$p < 0.05$. Vertical bars indicate ±*SE*. Data are slightly translated horizontally to increase the visibility of different conditions.

location. When a two-way ANOVA was performed, with SOA and saccade/reach location agreement as factors, the main effect of saccade/reach location agreement was not significant ($F(8,48) = 1.09$, $p > 0.30$), but interaction between the two factors was ($F(8,48) = 2.44$; $p < 0.05$). For the SOA 150-ms condition, reaches directed to the saccade location started after 488 ms ± 26 ms; these latencies were 63 ms shorter than when the reaches were directed to a different location (551 ms ± 42 ms; repeated measures *t*-test, $t(6) = -2.46$, $p < 0.05$); none of the other SOA conditions showed significant differences. The benefit observed at the SOA 150-ms condition could be explained by previous observations that people are faster to reach to the objects they are allowed to look at (Prablanc, Echallier, Komilis, & Jeannerod, 1979).

Finally, we analyzed movement errors. On trials when saccades and reaches were directed to different locations, participants made 23% of errors by looking at the location they were supposed to reach, and on 4% of trials they reached to the location they were supposed to look at. The proportion of errors did not vary as a function of SOA (ANOVA for saccade errors, with SOA as the main factor—$F(8,48) = 0.38$, $p > 0.9$; $F(8,48) = 0.99$, $p > 0.4$ for reach errors). This demonstrates that there is some crosstalk when saccade and reach targets have to be selected. It is not clear, though, whether these saccade and reach errors are due to participants being used to look and reach to the same locations in everyday situations.

In sum, these results show that dual-task costs arise for reaches when the saccade was executed first.

Experiment 3

In two parts of Experiment 3, we tested directly whether movement goal selection (in contrast to movement execution) is affected by the need to do two tasks simultaneously. As it has been shown that attention may shift to saccade and reach locations early during movement planning (Deubel & Schneider, 1996; Rizzolatti, Riggio, & Sheliga, 1994), we measured attention at saccade and reach locations by presenting an attentional probe—a briefly shown letter that participants had to report at the end of the trial. During this task, the first cue indicated a reach target and a subsequent cue indicated the saccade target (like in Experiment 1), and a probe could appear sometime during saccade or reach planning at different locations on the screen. If participants shifted their attention to saccade or reach location, probe discrimination should be better at those locations than at

other locations, to which no movement was planned. During the experiment, we also varied probe presentation time, which allowed us to determine at which point in time attention shifted to saccade or (and) reach locations. For example, it could be that attention deployment associated with saccade planning is delayed as long as the reaching does not start, leading to the prolongation of the saccade latencies as demonstrated in Experiment 1. Alternatively, it is possible that there are no dual-task costs to select a saccade goal even when the selection occurs during reach planning—this should be reflected in a parallel attention allocation to both saccade and reach targets before reaching onset.

Methods

Participants

Eight participants (mean age 23 years, 3 females) participated in Experiment 3. Ten participants took part in the "Saccade-only" control task (mean age 25, 4 females). They had normal or corrected-to-normal vision. Informed consent was obtained from each participant.

Apparatus and procedure

The apparatus and procedure of the experiment were the same as in Experiment 1, with the following exceptions. After the first movement cue appeared, participants had to reach to the object indicated by the cue. There were only 4 possible reaching locations (at 2, 5, 7, and 10 o'clock). With an SOA of 150 or 400 ms after the first cue, a second arrow cue was shown indicating the saccade goal (see Figure 1). The saccade goal could be located 3 or 5 items clockwise or anti-clockwise from the reaching location. Saccade and reaching movement goal selection was measured by presenting a probe stimulus. For this purpose, the display containing the mask elements changed into a display containing 11 distractor digits (digital "2" and "5") and one target character (digital letter "E" or digital "3"). This probe display was presented for 80 ms and was then masked. Given the short presentation time of the probe display, the probes could be detected only if participants attended to the probe location at the time when the probe was presented. The probe display could appear randomly in a time interval ranging from 50 ms before to 650 ms after the onset of the first movement cue. In other words, the mask elements could change into probe and distractors at any point of time, before the appearance of the first movement cue, up to a point in time when both movements were already finished. The probe was presented either at the saccade goal (33% of trials), at the reach goal (33% of trials), or at one of the other, movement-irrelevant locations (33% of trials). Participants were asked to indicate the probe identity ("E" or "3") at the end of each trial. We analyzed only trials where the probe appeared before eye movement onset.

Each block consisted of 144 trials. Participants completed at least 6 blocks of the task.

"Saccade-only" control experiment

In order to provide a baseline on how attention shifted to saccade goals when no simultaneous reach were to be made, we additionally performed a control experiment in which participants only looked at the object, without executing any reach movement (Saccade-only task). The design of this experiment was identical to Experiment 3, except that only one movement cue was presented. Participants had to saccade to the location indicated by the cue. The probe could be presented at the saccade target (50% of trials) or at a randomly selected, movement-irrelevant location (50% of trials). Each participant performed at least 4 experimental blocks of 192 trials each.

Results

As in the previous experiments, we observed dual-task costs when the planning processes for the two movements overlapped in time. Saccade latencies decreased with increasing SOA (316 ± 22 ms for SOA 150 ms as compared to 239 ± 13 ms for SOA 400 ms, repeated measures t-test, $t(7) = 6.53$, $p < 0.01$). On the other hand, reaching latencies were not affected by the SOA condition (286 ± 15 ms for SOA 150 ms and 286 ± 15 ms for SOA 400-ms conditions, repeated measures t-test, $p > 0.05$). Thus, saccade initiation was delayed if the reach was still being planned at the time of saccade cue presentation (SOA 150 ms). Saccade initiation was not delayed if the reach had already started, which was the case for the SOA 400-ms condition.

Next we analyzed whether participants were able to select movement goals during the preparation of the movements. For this purpose, we used probe discrimination rate as a measure of movement goal selection. Since the probe was presented at variable times, we were able to analyze the time course of attentional deployment to the probe locations. For each time point (every 50 ms), we calculated the proportion of trials in which participants correctly discriminated the probe. As we were interested in the shift of attention to saccade and reach goals before the movement onset, we excluded all trials in which probes were presented either after saccade or reach onset. The results are depicted in Figure 5. It can be seen that after the reach cue appeared, participants were at chance to discriminate the probes if they were presented at movement-irrelevant locations (probe discrimination was

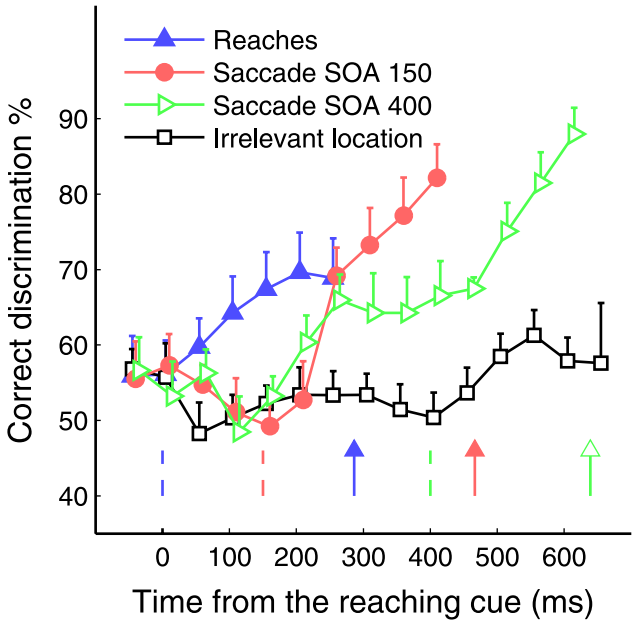


Figure 5. Probe discrimination rate at saccade and reach goals in the dual task of Experiment 3 (results do not include data from the Saccade-only task), as a function of time after reach cue presentation. Vertical dashed lines indicate the onsets of the cues for respective movements (e.g., blue dashed line—reaching cue presentation). Vertical arrows indicate the average movement latencies (e.g., blue arrow—reaching onset). Vertical error bars indicate ±*SE*. Data are slightly translated horizontally to increase the visibility of different conditions.

not different from chance level, $p > 0.05$). In contrast, probe discrimination at the reach goal increased gradually following the presentation of the respective cue. Further data analysis revealed that 50 ms after presentation of the reaching cue, participants became better than chance to discriminate probes presented at the reaching location (*t*-test comparing discrimination at reaching location versus 50% chance level, $t(7) = 2.56$, $p < 0.05$). This demonstrates that participants shifted their attention to the reach goal before the start of the reaching movement.

For the SOA 150-ms condition, and about 100 ms after the saccade cue appeared, probe discrimination became significantly better than chance also at the saccade goal ($t(7) = 5.13, p < 0.01$). After this point in time, i.e., already relatively long before saccade onset and also before the onset of the reach movement, participants were consistently better than chance to discriminate probes presented at the location of the saccade goal (all ps < 0.05). This shows that the attentional shift to the saccade goal started well before saccade onset. These results are in line with previous demonstrations showing increased probe discrimination at the saccade locations (Baldauf & Deubel, 2008; Deubel & Schneider, 1996; Godijn & Theeuwes, 2003).

Two important conclusions can be drawn from these results. First, improvement of probe discrimination at the saccade target was better than chance already before the reaching movement started. So, even though these saccades were markedly delayed due to the dual-task conditions, participants did not delay the *selection* of the saccade goal until after they started their reaching—the saccade goal was obviously selected before the start of the reaching movement. Second, the data demonstrate that attention was allocated to the two target locations simultaneously, as participants were better than chance to discriminate probes presented at both the saccade and the reaching goal before the reach started.

Somewhat unexpectedly, we found that for the SOA 400-ms condition, discrimination rate at the saccade goal increased already 150 ms before the saccade cue appeared ($t(7) = 2.88$, $p < 0.05$); from that time onward, participants were better than chance to discriminate probes presented at the saccade location. Note that after this initial increase in accuracy, discrimination rate at the saccade goal did not change over time until the appearance of the saccade cue. Only then, discrimination performance improved further. The predictive increase in probe discrimination accuracy suggests that participants tried to anticipate where they would have to make a saccade. If we assume that participants split their attention evenly between 4 possible saccade target locations, and given that probability to guess the probe identity correctly was 0.5 in our two-alternative forced choice task, then probe discrimination at possible saccade target should be 63% ($1/4 + (1 - 1/4) * 0.5$), which was similar to what we observed.

It should be noted here that, given the similarity of the initial increase of discrimination performance for both SOA conditions (red and green curves in Figure 5), we cannot exclude that anticipatory effects may also be involved in the SOA 150-ms condition. However, the assumption that the early attention shifts to the saccade target in the SOA 150-ms condition are elicited by the presentation of the saccade cue seems to be more parsimonious.

Further converging evidence for this assumption comes from the results of a parallel study in which we used a different combination of SOAs (SOA 150 ms and SOA 200 ms). In this study, we also observed that probe discrimination increased at the saccade location before the reach onset for the SOA 150-ms condition, while attention shifts were accordingly delayed for the SOA 200-ms condition (Jonikaitis & Deubel, in press, cf. Figure 4). Importantly, there were no anticipatory attentional shifts apparent for the SOA 200-ms condition in this study, which further confirms that saccade targets can indeed be selected during reach planning.

Thus combined we found that 250 ms after the reach cue appeared (time when green and red curves start rising in Figure 5)—and still before the reach onset—probe discrimination was already better than chance at either the already specified saccade goal (SOA 150-ms condition) or at the yet to be specified saccade goal (SOA 400-ms condition). These two observations strongly argue that

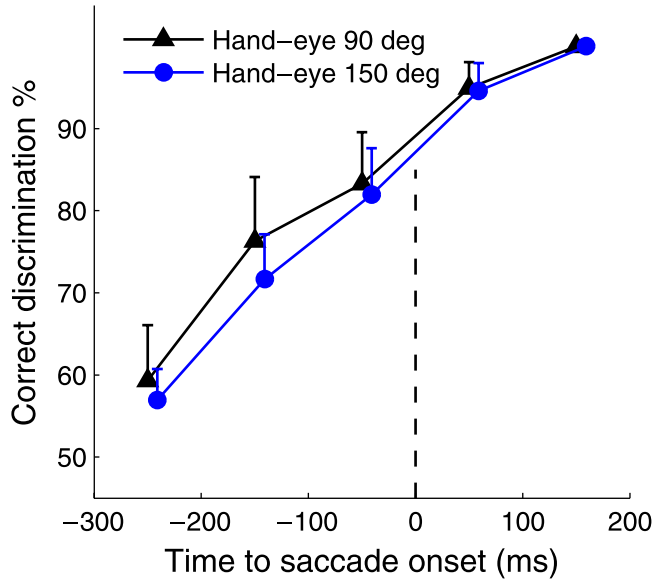


Figure 6. Probe discrimination at saccade goal in Experiment 3 as a function of the distance between eye and hand movement goal locations. Dashed gray line indicates saccade onset. Vertical error bars indicate ±*SE*. Data are slightly translated horizontally to increase the visibility of different conditions.

reach movement planning did not prevent the attentional shift to specified or potential saccade locations; thus, attentional selection of saccade locations was not delayed in time.

Even though participants shifted their attention to saccade locations during reach planning, it could be that this was an effect observed by mixing two different groups of trials in our design—on some trials, saccade and reach locations were close by, and on other trials, those locations were further away. Participants could have shifted their attention only to saccade locations further away or to saccade locations in the different visual hemifield than the reaches (for example, Alvarez & Cavanagh, 2005). To assess this possibility, we split the data by trials with saccades made to the opposite hemifield than the reaches and trials with saccades made to the same hemifield. We observed no effect on probe discrimination due to this data split (all $ps > 0.7$). We also split trials by whether saccade location was close or far from the reach location (3 or 5 items away from the reach object on the display). Again, we observed no discernible differences (all $ps > 0.5$; Figure 6).

Even though participants were able to select the saccade target before reaching onset, it is still possible that participants would have selected the target faster if there were no need to perform simultaneous reaching. In other words, the observed dual-task costs may have partly arisen because saccade target selection was somewhat delayed (even though it started before the reaching onset). To investigate this possibility, we compared the discrimination performance from the dual-task conditions with performance in the Saccade-only task, which did not include a reaching movement. Figure 7 shows discrimination performance, aligned to the time of saccade cue presentation, for the dual-task conditions and for the Saccade-only task. It can be seen that probe discrimination increased at about the same time after saccade cue onset in both the dual-task and the Saccade-only task. We calculated at which time probe discrimination after the saccade cue onset was better than performance 50 ms before the saccade cue onset (in order to equate for baseline differences in discrimination before cue onset in the SOA 400-ms task). This analysis shows that 100 ms after saccade cue onset in the Saccade-only task probe discrimination was better than baseline ($t(9) = 3.21$, $p < 0.05$); the same time value was found for the SOA 150-ms task ($t(7) = 4.86$, $p < 0.01$) and for the SOA 400-ms task ($t(7) = 2.85$, $p < 0.01$). Figure 7 includes also the data where the probe appeared after saccade onset. Note that in the Saccade-only task probe discrimination reached a certain level before the saccade and improved after saccade onset, as participants were then looking at the target directly. Interestingly, in the SOA 150-ms condition, probe discrimination at the saccade goal kept improving as long as the saccade did

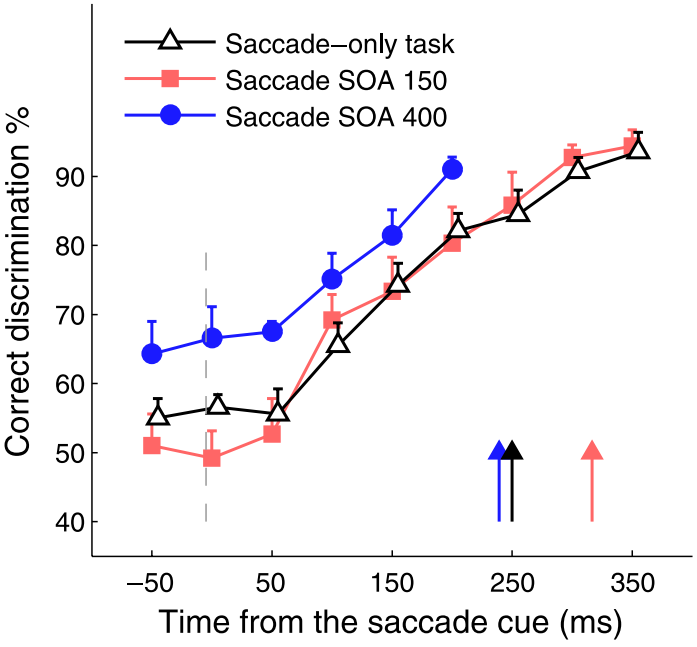


Figure 7. Probe discrimination at saccade goal during the dual task of Experiment 3 and the Saccade-only task. Dashed gray line indicates saccade cue presentation. Vertical arrows indicate average saccade latencies (i.e., saccade onset times) in the different conditions. Note that for both dual- and single-task conditions attention allocation to the saccade goal follows the same pattern, even after the onset of the saccade in the control Saccade-only task. Vertical error bars indicate ±*SE*. Data are slightly translated horizontally to increase the visibility of different conditions.

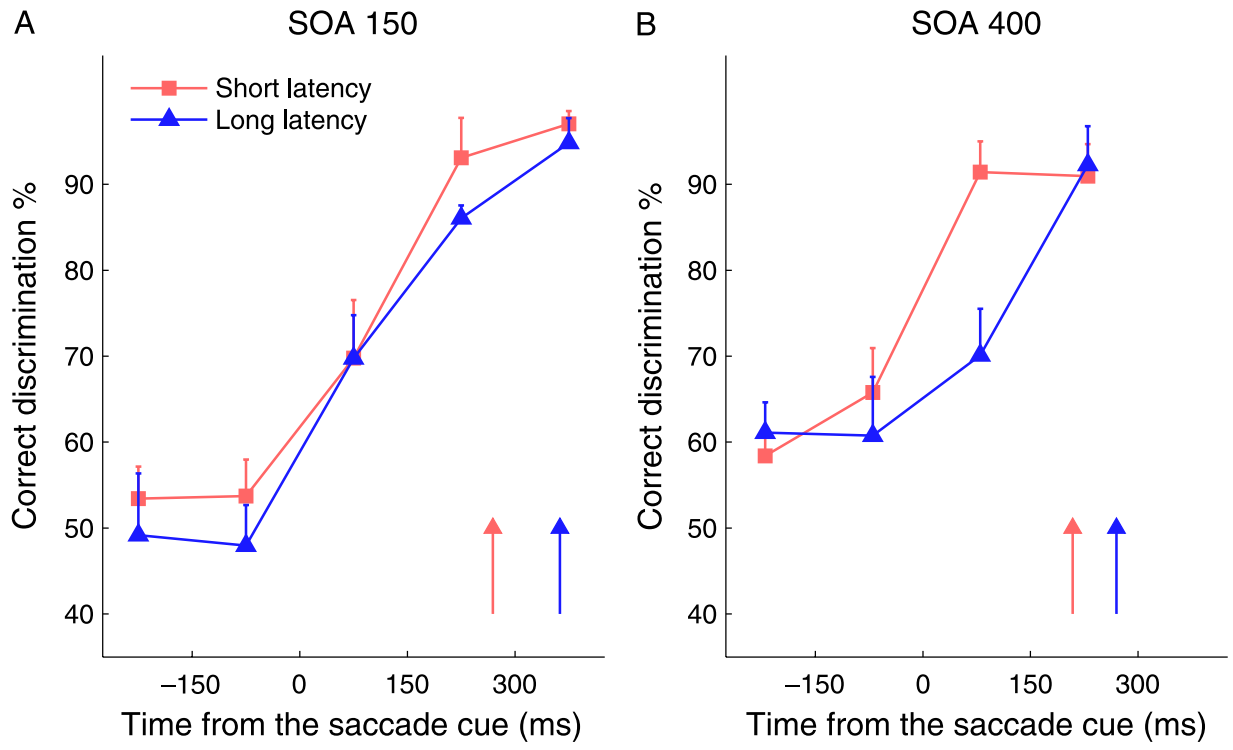


Figure 8. Probe discrimination before short and long latency saccades. Saccade cue appeared at time 0. Vertical arrows—average short and long saccade latencies. Color curves—probe discrimination rate at saccade target when saccades latencies were short (red line) or long (blue line). Vertical error bars indicate ±*SE*.

not start. The similarity of the temporal dynamics between the two conditions (SOA 150 ms and Saccade-only task) is striking, even though the saccade started in one condition considerably earlier than in the other. So, although saccade execution was considerably delayed in the SOA 150-ms condition, the attention shifts to the saccade target were not delayed. In other words, reach planning and execution did not delay saccade goal selection in the dual-task condition, as probe discrimination was not different from that in the Saccade-only task.

Further evidence that the delay of the saccades in the SOA 150-ms task was not related to the timing of the presaccadic attention shifts was provided by an analysis of the temporal relation between presaccadic attention shift and saccade onset. In a different study, we observed that saccades with shorter latencies are normally preceded by an earlier attention shift to the saccade target (Jonikaitis & Deubel, in press)—the faster the participants shifted their attention to the saccade goal, the shorter were the saccade latencies. For the dual-task conditions of the present experiment, we expected to find this tight temporal coupling between attention shift and saccade onset for the SOA 400-ms condition, where reaching movement and saccade processing no longer interfered. For the SOA 150-ms condition, however, the coupling should disappear, given that the dual-task costs as reflected in the saccade delay were unrelated to the presaccadic attention shift. In order to test this prediction, we split, for each SOA condition and each participant, saccade latencies by median into short latency saccades and long latency saccades.

As we had expected, saccadic reaction times were not related to speed of attention deployment in the SOA 150-ms condition. While the median split in this condition leads to a mean latency of 270 ± 16 ms for the faster saccades and 363 ± 29 ms for the slower saccades, this difference is not reflected in the attentional allocation for longer or shorter latency saccades (Figure 8, left panel, all repeated measures comparisons not significant).

In the SOA 400-ms condition, the trials with faster saccades had an average saccade latency of 209 ± 16 ms; the trials with slower saccades had an average latency of 270 ± 17 ms. As can be seen from the right graph of Figure 8, probe discrimination at the saccade goal increased earlier for the trials when saccades had shorter latencies than on trials with longer latency saccades. At 150 ms after the saccade cue, discrimination was better at the saccade target if the saccade latencies were shorter ($t(7) = 2.56$, $p < 0.05$), while attention deployment occurred considerably later for the slower saccades. Thus, at the time when there were no dual-task costs observed, earlier attention shifts were associated with shorter saccade latencies. This finding suggests that the dual-task

costs in the SOA 150-ms condition arise at a processing stage different from—and probably later than—the attentional selection of the saccade goal.

Together the findings show that the participants were able to shift their attention to saccade and reaching goals before reaching onset, and that there was no delay in saccade goal selection while the reaching was planned. Thus, while the saccade onset became markedly delayed due to the dual-task condition, this delay was not reflected in the time course of attentional allocation before the saccade.

Discussion

Our experiments investigated whether dual-task costs in the simultaneous planning of eye and hand movements result from a competition for attentional resources. Movement latencies showed large dual-task costs when saccades had to be planned during reaching preparation. However, these costs did not arise from the attentional selection of the movement goals. The results show that participants can shift their attention to a saccade target even while the reaching movement is being planned and has not yet started.

Dual-task costs in the planning of saccades and reaches

We found that there were large dual-task costs when the planning of goal-directed saccades and reaches overlapped in time. Our results are comparable to findings reported by Pashler et al. (1993). In their study, participants were not able to elicit a saccade if the central cue instructing the saccade appeared while the participants performed a tone discrimination task requiring a manual button press. The magnitude of the effects found in our study (about 100-ms dual-task cost for saccades made in the SOA 150-ms condition) was about equal to the effect observed in Pashler et al.'s study (also 100-ms cost for SOA 150 ms). The main difference between these two studies is that in our experiments participants had to plan two movements directed to different locations, whereas in the study of Pashler et al. the first task was a button press and the second task was a goal-directed saccade.

The dual-task interference observed in our experiments could result from various stages of movement planning. In our task, participants had to interpret each cue, select an appropriate response (to make an eye or a hand movement), and to plan the movement itself. Movement planning consists of selecting an appropriate target for the movement and specifying all movement parameters. Additionally, factors such as uncertainty about when the second cue appears (Gottsdanker, 1980), impaired timing judgments during dual tasks (Brown, 1997), and confusability regarding the direction of motion of the effector (Huestegge & Koch, 2009) may also play a role. Our results suggest that one of the most important processes involved in the task, namely the selection of the movement goals, did not cause the dual-task interference. It remains to be investigated at which stage during movement planning the interference actually occurs.

It is difficult to directly compare our results to some of the other studies that investigated saccade-reaching dual-task costs, since these did not systematically manipulate the overlap between saccade and reach planning (Bekkering et al., 1994, 1995; Lünenburger et al., 2000). Although it has been reported that saccade latencies are shorter if concurrent reaches to the same object are planned (Lünenburger et al., 2000), the opposite pattern of results was found in a different set of studies (Bekkering et al., 1994, 1995). It is possible that the requirement to make two movements to the same object simultaneously might evoke a pattern of eye–hand coordination that is "hard-wired." For example, both Lünenburger et al. and Bekkering et al. have suggested that the superior colliculus might mediate the observed coupling between the eye and hand, as some of the neurons in intermediate and deep layers of superior colliculus are known to fire before arm movements (Werner, Dannenberg, & Hoffmann, 1997). The assumption that simultaneous eye and hand movements might be coordinated in a special way is also supported by the finding that saccade durations decrease if saccades are made simultaneously with hand movements (Snyder, Calton, Dickinson, & Lawrence, 2002).

Movement goal selection for eye and hand

Another matter of debate in eye–hand movement studies has been whether the target representation for movement planning is shared between both systems or is separate. We did not find a saccadic latency benefit when the saccade was planned to the same location as the reach. Thus, even though participants selected a target for the reach, they were not faster to saccade to that same target than to saccade to a different target. This indicates that movement goal selection for the eye and the hand movements is relatively independent. In other words, when the reach is planned, the saccade does not have to be planned to the same target (for a similar observation, see also Stritzke & Trommershäuser, 2007).

Our results argue against some findings that were interpreted as showing that eye and hand movement goal selection is shared. Neggers and Bekkering (2000, 2001), for example, reported that if participants are making a saccade and a reach to the same object, they are unable to move their eyes to a different location while the hand is still moving. In addition, it has been observed that saccade latencies are longer if a simultaneous hand movement is

planned to the same location (Bekkering et al., 1994, 1995).

We think that those studies could be interpreted in a different way—it might be advantageous to keep the eyes stable while the hand movement is planned or executed. A number of studies, behavioral and neurophysiological, show that eye position influences the planning for reaching and pointing (Batista, Buneo, Snyder, & Andersen, 1999; Medendorp & Crawford, 2002; Medendorp, Goltz, Vilis, & Crawford, 2003). This indicates that the visual system keeps track of where the hand and the reach goals are relative to the eye and suggests that every eye movement requires the recalculation of the hand movement goal position with respect to the new eye position. Thus, keeping the eyes stable might be advantageous for fast hand movement planning, but this coupling does not necessarily mean that movement goal selection is shared for eye and hand movements. Further research needs to be carried out to clarify whether targets for eye and hand are selected independently.

In Experiment 3, we demonstrated that two targets, one for the saccade and one for the reach, can be selected in parallel, before reaching movement onset. In other words, before reaches started, participants were attending simultaneously to both saccade and reach locations. In addition, attention was allocated to the saccade goal immediately after the saccade cue onset—regardless of SOA. Thus, it did not matter whether the hand movement was planned at that time or not—participants selected the saccade target immediately after saccade cue onset. This demonstrates that saccade goal selection was independent of whether the reach goal was selected at that time or not. The finding further supports the conjecture that the mechanisms selecting the goals for eye and hand movements are dynamically independent (Jonikaitis & Deubel, in press).

Split attention

We also demonstrate that attention can be split to multiple locations, as illustrated by our finding that probe discrimination was better than chance at saccade and reach goal locations before reach onset. That attention can be split has been proposed in a number of studies (e.g., Adamo, Pun, Pratt, & Ferber, 2008; Alvarez & Cavanagh, 2005; Awh & Pashler, 2000; Bichot, Cave, & Pashler, 1999); however, this view has also been vigorously objected (e.g., Dubois, Hamker, & VaRullen, 2009; Jans, Peters, & Weerd, 2010). Our data clearly support the view that attention can be split to parallel locations in a task involving the preparation of eye and hand movements, in line with further recent evidence (Jonikaitis & Deubel, in press). One interesting question concerns how this split is achieved. Our task, contrary to typical tasks investigating parallel attention foci, did not explicitly instruct attention to shift to any location. The main task was the movement task, and we observed that probe discrimination increased at the movement goal locations. The shift of attention to the movement goals seems to be involuntary to some degree, as probe discrimination at movement goal locations increases even when participants are explicitly informed that probe is more likely to appear at other locations (Deubel & Schneider, 1996; Jonikaitis & Deubel, in press; Tibber, Grant, & Morgan, 2009; Wilder, Kowler, Schnitzer, Gersch, & Dosher, 2009). This seems to be true also in cases where no discrimination task is present, but attention is measured using ERPs (Baldauf & Deubel, 2009). Moreover, attention was found to shift to multiple locations when a sequence of eye or hand movements to multiple targets is prepared (Baldauf & Deubel, 2008, 2010; Baldauf, Wolf, & Deubel, 2006; Godijn & Theeuwes, 2003). All these evidences suggest that attentional resources can be distributed to multiple targets during the planning of combined eye and hand movements as shown here, as well as during the preparation of movement sequences.

The question still remains as to the relationship between automatic attention allocation to movement goals as studied here and the intentional, simultaneous attention allocation to multiple stimuli. It could be that different attentional resources exist for the shifting attention before movement onset and the intentional attending to other locations (Montagnini & Castet, 2007). While this question remains to be investigated, our data support the view that attention can be transiently split.

Inhibition of return

We also observed that saccades were delayed when participants already reached to that location. This effect occurred late, at an SOA of around 700–900 ms and thus was within the time frame when Inhibition of Return (IOR) is known to occur (Klein, 1988, 2000). IOR is regarded as a mechanism that discourages attentional (or saccadic) revisiting of previously attended locations. Our results show that targets selected for hand movements can inhibit saccadic orienting to those targets. In other words, within the IOR time frame, participants tended not to direct saccades to the locations they already reached at.

It has been suggested that IOR originates from either attentional or saccadic systems. A possible attentional explanation of our findings is that participants shifted their attention to the hand movement target when they planned the hand movement. Later, when the saccade had to be planned to that same target, the shift of attention to this location was delayed, resulting in the observed IOR effect.

Another possible explanation is that the observed IOR is a saccadic effect (Theeuwes & Godijn, 2002). It could be argued that participants planned a saccade to every reach target—without executing the saccade, which resulted in an IOR effect. However, if this were the case, then at short SOAs saccades directed to the reaching goal should have been faster than saccades directed elsewhere, a result that

we did not observe (see Figure 2). Our findings thus argue for an attentional origin of IOR.

Reaction time is not attention

A striking observation of this study is that while saccades showed large dual-task costs as measured in saccadic latencies, there were no attentional target selection costs, i.e., the attention shift preceding the saccade showed no delay. This is surprising given the common assumption that attention and saccades are closely coupled when people are asked to make speeded responses while eye or hand movements are planned. The clear dissociation between saccadic reaction time and attentional selection indicates that caution should be taken in using saccade or hand movement latencies as a measure of target selection or attentional allocation. Instead of attentional processing, the latencies may merely reflect dual-task constraints occurring at later stages of sensorimotor processing.

Acknowledgments

This study was supported by the Deutsche Forschungsgemeinschaft (GRK 1091) and by the 7th Framework Program of the European Community (Project "GRASP," ICT-215821).

Commercial relationships: none.
Corresponding author: Donatas Jonikaitis.
Email: djonikaitis@gmail.com.
Address: Allgemeine und Experimentelle Psychologie, Ludwig-Maximilians-Universität, München, Germany.

References

Adamo, M., Pun, C., Pratt, J., & Ferber, S. (2008). Your divided attention, please! The maintenance of multiple attentional control sets over distinct regions in space. *Cognition, 107,* 295–303. [PubMed]

Alvarez, G. A., & Cavanagh, P. (2005). Independent resources for attentional tracking in the left and right visual hemifields. *Psychological Science, 16,* 637–643. [PubMed]

Andersen, R. A., & Buneo, C. A. (2002). Intentional maps in posterior parietal cortex. *Annual Review of Neuroscience, 25,* 189–220. [PubMed]

Awh, E., & Pashler, H. (2000). Evidence for split attentional foci. *Journal of Experimental Psychology: Human Perception and Performance, 26,* 834–846. [PubMed]

Baldauf, D., & Deubel, H. (2008). Properties of attentional selection during the preparation of sequential saccades. *Experimental Brain Research, 184,* 411–425. [PubMed]

Baldauf, D., & Deubel, H. (2009). Attentional selection of multiple goal positions before rapid hand movement sequences: An event-related potential study. *Journal of Cognitive Neuroscience, 21,* 18–29. [PubMed]

Baldauf, D., & Deubel, H. (2010). Attentional landscapes in reaching and grasping. *Vision Research, 50,* 999–1013. [PubMed]

Baldauf, D., Wolf, M., & Deubel, H. (2006). Deployment of visual attention before sequences of goal-directed hand movements. *Vision Research, 46,* 4355–4374. [PubMed]

Batista, A. P., Buneo, C. H., Snyder, L. H., & Andersen, R. A. (1999). Reach plans in eye-centered coordinates. *Science, 285,* 257–260. [PubMed]

Bekkering, H., Adam, J. J., Kingma, H., Huson, A., & Whiting, H. T. (1994). Reaction time latencies of eye and hand movements in single- and dual-task conditions. *Experimental Brain Research, 97,* 471–476. [PubMed]

Bekkering, H., Adam, J. J., van den Aarssen, A., Kingma, H., & Whiting H. T. (1995). Interference between saccadic eye and goal-directed hand movements. *Experimental Brain Research, 106,* 475–484. [PubMed]

Bichot, N. P., Cave, K. R., & Pashler, H. (1999). Visual selection mediated by location: Feature based selection of noncontiguous locations. *Perception & Psychophysics, 61,* 403–423. [PubMed]

Brown, S. W. (1997). Attentional resources in timing: Interference effects in concurrent temporal and nontemporal working memory tasks. *Perception & Psychophysics, 59,* 1118–1140. [PubMed]

Carrasco, M. (2006). Covert attention increases contrast sensitivity: Psychophysical, neurophysiological and neuroimaging studies. *Progress in Brain Research, 154,* 33–70. [PubMed]

Deubel, H., & Schneider, W. X. (1996). Saccade target selection and object recognition: Evidence for a common attentional mechanism. *Vision Research, 36,* 1827–1837. [PubMed]

Deubel, H., Schneider, W. X., & Paprotta, I. (1998). Selective dorsal and ventral processing: Evidence for a common attentional mechanism in reaching and perception. *Visual Cognition, 5,* 81–107.

Dubois, J., Hamker, F. H., & VanRullen, R. (2009). Attentional selection of noncontiguous locations: The spotlight is only transiently "split". *Journal of Vision, 9*(5):3, 1–11, http://www.journalofvision.org/content/9/5/3, doi:10.1167/9.5.3. [PubMed] [Article]

Godijn, R., & Theeuwes, J. (2003). Parallel allocation of attention prior to the execution of saccade sequences. *Journal of Experimental Psychology: Human Perception and Performance, 29,* 882–896. [PubMed]

Gottsdanker, R. (1980). The ubiquitous role of preparation. In G. E. Stelmach & J. Requin (Eds.), *Tutorials in motor behavior* (pp. 315–371). Amsterdam, The Netherlands: North-Holland.

Hommel, B. (1998). Automatic stimulus-response translation in dual-task performance. *Journal of Experimental Psychology: Human Perception and Performance, 24,* 1368–1384. [PubMed]

Horstmann, A., & Hoffmann, K. P. (2005). Target selection in eye–hand coordination: Do we reach to where we look or do we look to where we reach? *Experimental Brain Research, 167,* 187–195. [PubMed]

Huestegge, L., & Koch, I. (2009). Dual-task crosstalk between saccades and manual responses. *Journal of Experimental Psychology: Human Perception and Performance, 35,* 352–362. [PubMed]

Jans, B., Peters, J. C., & De Weerd, P. (2010). Visual spatial attention to multiple locations at once: The jury is still out. *Psychological Review, 11,* 637–684. [PubMed]

Johansson, R. S., Westling, G., Backstrom, A., & Flanagan, J. R. (2001). Eye–hand coordination in object manipulation. *Journal of Neuroscience, 21,* 6917–6932. [PubMed]

Jonikaitis, D., & Deubel, H. (in press). Independent allocation of attention to eye and hand targets in coordinated eye–hand movements. *Psychological Science*.

Kahneman, D. (1973). *Attention and effort*. Englewood Cliffs, NJ: Prentice Hall.

Klein, R. (1988). Inhibitory tagging system facilitates visual search. *Nature, 334,* 430–431. [PubMed]

Klein, R. (2000). Inhibition of return. *Trends in Cognitive Sciences, 4,* 138–147. [PubMed]

Land, M. F., & Hayhoe, M. M. (2001). In what ways do eye movements contribute to everyday activities? *Vision Research, 41,* 3559–3565. [PubMed]

Lien, M.-C., & Proctor, R. W. (2002). Stimulus-response compatibility and psychological refractory period effects: Implications for response selection. *Psychonomic Bulletin & Review, 9,* 212–238. [PubMed]

Linnell, K. J., Humphreys, G. W., McIntyre, D. B., Laitinen, S., & Wing, A. M. (2005). Action modulates object-based selection. *Vision Research, 45,* 2268–2286. [PubMed]

Lünenburger, L., Kutz, D. F., & Hoffmann, K. P. (2000). Influence of arm movements on saccades in humans. *European Journal of Neuroscience, 12,* 4107–4116. [PubMed]

Medendorp, W. P., & Crawford, J. D. (2002). Visuospatial updating of reaching targets in near and far space. *NeuroReport, 13,* 633–636. [PubMed]

Medendorp, W. P., Goltz, H. C., Vilis, T., & Crawford, J. D. (2003). Gaze-centered updating of visual space in human parietal cortex. *Journal of Neuroscience, 23,* 6209–6214. [PubMed]

Milner, A. D., & Goodale, M. A. (1995). *The visual brain in action*. Oxford, United Kingdom: Oxford University Press.

Montagnini, A., & Castet, E. (2007). Spatiotemporal dynamics of visual attention during saccade preparation: Independence and coupling between attention and movement planning. *Journal of Vision, 7*(14):8, 1–16, http://www.journalofvision.org/content/7/14/8, doi:10.1167/7.14.8. [PubMed] [Article]

Neggers, S. W. F., & Bekkering, H. (2000). Ocular gaze is anchored to the target of an ongoing pointing movement. *Journal of Neurophysiology, 83,* 639–651. [PubMed]

Neggers, S. W. F., & Bekkering, H. (2001). Gaze anchoring to a pointing target is present during the entire pointing movement and is driven by a non-visual signal. *Journal of Neurophysiology, 86,* 961–970. [PubMed]

Niechwiej-Szwedo, N., McIlroy, W. E., Green, R., & Verrier M. C. (2005). The effect of directional compatibility on the response latencies of ocular and manual movements. *Experimental Brain Research, 162,* 220–229. [PubMed]

Pashler, H. E. (1994). Dual-task interference in simple tasks: Data and theory. *Psychological Bulletin, 116,* 220–244. [PubMed]

Pashler, H. E., Carrier, M., & Hoffman, J. (1993). Saccadic eye movements and dual-task interference. *Quarterly Journal of Experimental Psychology, 46,* 51–82. [PubMed]

Pelz, J., Hayhoe, M. M., & Loeber, R. (2001). The coordination of eye, head, and hand movements in a natural task. *Experimental Brain Research, 139,* 266–277. [PubMed]

Prablanc, C., Echallier, J. F., Komilis, E., & Jeannerod, M. (1979). Optimal response of eye and hand motor systems in pointing at visual target. *Biological Cybernetics, 35,* 113–124. [PubMed]

Rizzolatti, G., Riggio, L., & Sheliga, B. M. (1994). Space and selective attention. In C. Umilta & M. Moscovitch (Eds.), *Attention and performance XV: Conscious and nonconscious information processing* (pp. 231–265). Cambridge, MA: MIT Press.

Schiegg, A., Deubel, H., & Schneider, W. X. (2003). Attentional selection during preparation of prehension movements. *Visual Cognition, 10,* 409–431.

Schubert, T. (1999). Processing differences between simple and choice reactions affect bottleneck localization in overlapping tasks. *Journal of Experimental Psychology: Human Perception and Performance, 25,* 1–18.

Schubert, T. (2008). The central attentional limitation and executive control. *Frontiers in Bioscience, 13,* 3569–3580. [PubMed]

Sharikadze, M., Cong, D. K., Staude, G., Deubel, H., & Wolf, W. (2008). Dual-tasking: Is manual tapping independent of concurrently executed saccades? *Brain Research, 1283,* 41–49. [PubMed]

Snyder, L. H., Calton, J. L., Dickinson, A. R., & Lawrence, B. M. (2002). Eye–hand coordination: Saccades are faster when accompanied by a coordinated arm movement. *Journal of Neurophysiology, 87,* 2279–2286. [PubMed]

Song, J.-H., & Nakayama, K. (2006). Role of focal attention on latencies and trajectories of visually guided manual pointing. *Journal of Vision, 6*(9):11, 982–995, http://www.journalofvision.org/content/6/9/11, doi:10.1167/6.9.11. [PubMed] [Article]

Stritzke, M., & Trommershäuser, J. (2007). Eye movements during rapid pointing under risk. *Vision Research, 47,* 2000–2009. [PubMed]

Theeuwes, J., & Godijn, R. (2002). Oculomotor capture and inhibition of return: Evidence for an oculomotor suppression account of IOR. *Psychological Research, 66,* 234–246. [PubMed]

Tibber, M. S., Grant, S., & Morgan, M. J. (2009). Oculomotor responses and visuospatial perceptual judgments compete for common limited resources. *Journal of Vision, 9*(12):21, 1–13, http://www.journalofvision.org/content/9/12/21, doi:10.1167/9.12.21. [PubMed] [Article]

Werner, W., Dannenberg, S., & Hoffmann, K. P. (1997). Arm-movement-related neurons in the primate superior colliculus and underlying reticular formation: Comparison of neuronal activity with EMGs of muscles of the shoulder, arm and trunk during reaching. *Experimental Brain Research, 115,* 191–205. [PubMed]

Wilder, J. D., Kowler, E., Schnitzer, B. S., Gersch, T. M., & Dosher, B. A. (2009). Attention during active visual tasks: Counting, pointing, or simply looking. *Vision Research, 49,* 1017–1031. [PubMed]

Independent Allocation of Attention to Eye and Hand Targets in Coordinated Eye-Hand Movements

Donatas Jonikaitis and Heiner Deubel

Psychological Science published online 26 January 2011
DOI: 10.1177/0956797610397666

Research Article

Psychological Science
XX(X) 1–9

DOI: 10.1177/0956797610397666
http://pss.sagepub.com

Independent Allocation of Attention to Eye and Hand Targets in Coordinated Eye-Hand Movements

Donatas Jonikaitis and Heiner Deubel
Ludwig-Maximilians-Universität

Abstract

When reaching for objects, people frequently look where they reach. This raises the question of whether the targets for the eye and hand in concurrent eye and hand movements are selected by a unitary attentional system or by independent mechanisms. We used the deployment of visual attention as an index of the selection of movement targets and asked observers to reach and look to either the same location or separate locations. Results show that during the preparation of coordinated movements, attention is allocated in parallel to the targets of a saccade and a reaching movement. Attentional allocations for the two movements interact synergistically when both are directed to a common goal. Delaying the eye movement delays the attentional shift to the saccade target while leaving attentional deployment to the reach target unaffected. Our findings demonstrate that attentional resources are allocated independently to the targets of eye and hand movements and suggest that the goals for these effectors are selected by separate attentional mechanisms.

Keywords

attention, saccades, reaching, hand movements

Received 3/7/10; Revision accepted 9/9/10

To interact with objects in their environment, humans often make a combination of eye movements and reaching movements. The control of these movements is not independent: Typically, the eye movement pattern is organized such that it helps to gather the information that is most important for reaching and manipulating an object. For instance, in a "pick and place" task, humans fixate the item to be picked up, look at possible obstacle locations when lifting it, and then make a saccade to the end goal of the hand movement before the hand reaches this location (Horstmann & Hoffmann, 2005; Johansson, Westling, Backstrom, & Flanagan, 2001; Land, Mennie, & Rusted, 1999; Pelz, Hayhoe & Loeber, 2001).

Planning these coordinated movements requires the selection of the movement targets. Given the commonly observed yoking of eye and hand movements, the question arises whether the targets for the eye and the targets for the hand are selected by a common mechanism or by independent systems. The first possibility would indicate that planning coordinated eye-hand movements is based on selecting a common target and results in eye-hand coupling at early stages of movement planning. Alternatively, if selecting targets for eye and hand movements involves separate, largely independent systems, eye-hand coordination may occur only at later stages of sensorimotor processing.

The view that the selection of goals for eye movements and hand movements involves separate, largely independent systems has gained wide support from a number of neurophysiological studies. Single-cell recording studies in monkeys have suggested that separate areas in parietal cortex represent movement goals for saccades and for reaches (Calton, Dickinson, & Snyder, 2002; Dickinson, Calton, & Snyder, 2003; Snyder, Batista, & Andersen, 1997). Functional imaging and magnetoencephalography studies in humans have identified distinct parietal regions that show preparatory activity before eye or hand movements begin (Tosoni, Galati, Romani, & Corbetta, 2008; Van Der Werf, Jensen, Fries, & Medendorp, 2010). Activity associated with selection of saccade goals has also been demonstrated in prefrontal cortex, in both single-cell and neuroimaging studies (e.g., Curtis & Connolly, 2008; Lawrence & Snyder, 2009). Finally, findings from psychophysical studies have revealed that the early stages of movement planning are separate for eye movements and hand movements (Prablanc, Echallier, Komilis, & Jeannerod, 1979;

Corresponding Author:
Donatas Jonikaitis, Allgemeine und Experimentelle Psychologie, Ludwig-Maximilians-Universität, Leopoldstr. 13, Munich 80802, Germany
E-mail: djonikaitis@gmail.com

Sailer, Eggert, Ditterich, & Straube, 2000; Thompson & Westwood, 2007).

However, the alternative view that a single system underlies the selection of goals for eye and hand movements has also found support in a number of psychophysical studies (Bekkering, Adam, van den Aarssen, Kingma, & Whiting, 1995; Neggers & Bekkering, 2000; Song & McPeek, 2009). Further evidence for this view has come from functional imaging studies showing an overlap of the systems involved in selecting eye and hand movements in both parietal and prefrontal cortex (Beurze, de Lange, Toni, & Medendorp, 2009; Levy, Schluppeck, Heeger, & Glimcher, 2007).

It is important to note that the psychophysical studies we mentioned used measures related to motor output, such as correlations between precision of saccade and reaching endpoints, movement velocity profiles, and movement latencies. Therefore, the results cannot speak directly to the issue of whether the coupling of eye and hand occurs at the early stages of movement planning involving selection of movement targets or at later processing stages. Here, we report four experiments in which we studied the selection of movement goals directly by using spatial attention as an index of target selection. We based our approach on the well-established fact that visual attention is allocated to the target of the planned movement before saccades (Deubel & Schneider, 1996; Kowler, Anderson, Dosher, & Blaser, 1995; Montagnini & Castet, 2007) and reaching movements (Deubel, Schneider, & Paprotta, 1998; Linnell, Humphreys, McIntyre, Laitinen, & Wing, 2005) occur. Perceptual measures of attentional allocation are therefore direct indicators of movement goal selection. In our experiments, we asked participants to make a saccade and a reach to spatially separate targets while their allocation of attention was measured by a probe-discrimination task. The results show that selection of the saccade target and the reach goal can occur independently, suggesting that the goals of eye and hand movements are selected by separate mechanisms. This also implies that eye-hand coupling does not result from a common attentional selection mechanism, but probably follows from interactions at later processing stages.

Experiment 1

In this experiment, we established that attention is allocated to movement-goal locations before movement onset. Participants either made a saccade to a centrally cued target (saccade-only task) or reached toward the cued target without looking at it (reach-only task). We measured covert attentional allocation by comparing probe-discrimination rates for probes at movement goals and probes at movement-irrelevant locations.

Method

Participants sat in a dimly illuminated room with their right hand on an inclined surface and under a half-translucent mirror (which reflects light coming from above it and lets through light coming from below it). Stimuli were projected onto the mirror from a monitor above it. This setup caused the projected visual stimuli to appear on the mirror and prevented participants from seeing the reaching hand below the mirror. Visual feedback about the accuracy of reaching movements was provided by an LED that was fixed to a fingertip and could be switched on and off during the experiment. Switching on the LED resulted in participants being able to see where their reaching movement ended with respect to the movement target shown on the mirror. Reaching movements were recorded at 120 Hz with a Fastrack (Polhemus Inc., Colchester, VT) electromagnetic position sensor attached to the index finger of the right hand. Eye movements were recorded with a video-based eye-tracking system (Eyelink-I, SensoMotoric Instruments, Teltow, Germany) at a temporal resolution of 250 Hz.

Figure 1 depicts the stimulus sequence. Twelve mask elements (size: 0.9° × 1.4°; composed of randomly generated lines) were presented on a uniform gray background and arranged on an imaginary circle with a radius of 6.5°. Participants first directed their index finger and their gaze to a central fixation cross. At a time between 580 and 880 ms after fixation, the central cross changed into an arrow that pointed toward any 1 of the 12 mask stimuli. Participants either made a saccade toward the cued location (saccade-only task) or reached to the cued location while maintaining central fixation (reach-only task). Visual feedback about reaching accuracy was given 1,500 ms later.

While performing the saccade task or reaching task, participants had to detect a brief probe stimulus that was shown at one

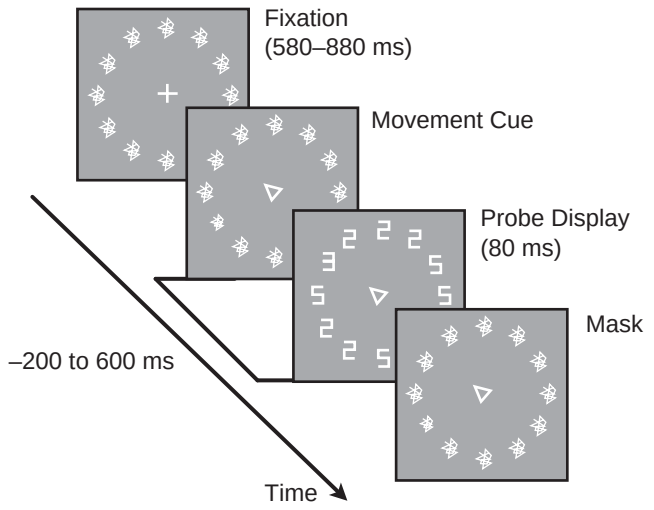


Fig. 1. The stimulus sequence for Experiments 1 and 2. Each trial began with 12 mask elements displayed in a circle. In Experiment 1, participants quickly looked (saccade-only task) or reached to (reach-only task) the item indicated by the centrally presented movement cue (arrow). A probe display comprising 11 distractors (digital *2*s and *5*s) and the probe letter (a digital *E* or *3*) appeared between 200 ms before and 600 ms after the onset of the movement cue, and participants reported the identity of the probe. After 80 ms, the probe was masked. In Experiment 2, the movement cue was the signal to initiate both a reach to the location indicated by the cue and a saccade to a prespecified location that was constant within a given experimental block.

of the locations initially occupied by the mask elements (i.e., either the movement-goal location or one of the movement-irrelevant locations). At a random time between 200 ms before and 600 ms after the onset of the movement cue, 11 of the 12 mask stimuli changed into distractors (digital *2*s and *5*s), while 1 mask stimulus changed into the probe letter (a digital *E* or *3*). The probe display was presented for 80 ms, and then the 12 masks reappeared. After finishing the eye or hand movement, participants reported whether they had perceived an *E* or a *3*. Responses were made by nonspeeded button presses with the left hand. The probe appeared at the movement-goal location with 50% probability and elsewhere (a movement-irrelevant location) on the other 50% of trials; the probe never appeared directly beside the movement-goal location.

A total of 10 observers took part in the saccade-only task, and 11 observers took part in the reach-only task. Each participant performed at least four experimental blocks of 192 trials each.

Results

Because the probe was presented at variable times within the experimental sequence, we were able to analyze the time course of attentional deployment to the probe locations. For each time point (every 30 ms), we calculated the proportion of probe discriminations that were correct. The probe-discrimination rate for probes located at the movement goal, whether a saccade goal or a reach goal, increased gradually over time before the movement onset (Fig. 2a). In the saccade-only task, when the probe appeared at the saccade target, discrimination performance improved to a level significantly above chance at around 80 ms after movement-cue onset, $t(9) = 3.30$, $p < .01$; after that time, probe discrimination was always better than chance, all ps $< .05$. In the reach-only task, when the probe was at the reach goal, discrimination performance improved to a level significantly above chance at around 140 ms after movement-cue onset, $t(10) = 3.25$, $p = .01$, and was better than chance for all times afterward. Immediately before movement onset (mean saccade latency = 250 ms, *SEM* = 6 ms; mean reach latency = 295 ms, *SEM* = 12 ms), probe-discrimination rates for probes at the saccade and reach goals were comparable, $p > .05$. Participants performed at chance levels in discriminating probes at movement-irrelevant locations.

These findings demonstrate that before a saccade or reach started, attention shifted to the location of the movement goal, resulting in above-chance probe discrimination at that location. In contrast, participants performed at chance levels when reporting the identity of a probe at a location to which no action was directed.

Experiment 2

In this experiment, participants performed a combined-movement task, making simultaneous eye and hand movements to two separate locations (except for a few trials in which both movements were directed to the same location). Again, we measured attentional allocation by having participants report the identity of a probe.

Method

In this task, saccade target was kept constant (at the clock position of 2, 4, 8, or 10 o'clock) for each block of 190 trials, and before each block, participants were instructed verbally about the location of the saccade target. Stimuli and timing were the same as in Experiment 1. When the movement cue appeared, participants were asked to make two movements simultaneously: a reach to the cued location and a saccade to the remembered location. We used a fixed saccade target because even when a saccade can be prepared for a known location, the attention shift to the target is obligatory (Deubel & Schneider, 2003). We used four different spatial distances between the saccade and the reaching goals: no elements (saccade and reach directed to the same location), one element (reach target next to the saccade target), three elements, or five elements. The probe appeared at the saccade target on 33% of the trials, at the reach target on 33% of the trials, and at one of the other (movement-irrelevant) locations on 33% of the trials. Ten observers participated in the study. Each participant performed at least six experimental blocks, covering each saccade target location at least once.

Results

Mean saccade latency in this task was 288 ms (*SEM* = 16 ms); mean reach latency was 300 ms (*SEM* = 20 ms). Figure 2b shows participants' discrimination performance for probes appearing at various times before and after saccade onset. It is striking that probe-discrimination performance increased for probes at the saccade target and probes at the reach target at about the same time relative to saccade onset. Indeed, before the saccades started, the probe-discrimination rates were comparable for probes at the saccade target and probes at the reach target at each time point, all ps $> .05$ (repeated measures t tests), which indicates that attention was allocated to the two movement goals in parallel. In contrast, participants performed at chance levels in discriminating probes at movement-irrelevant locations.

Having found evidence for the parallel allocation of attention to the two movement targets, we next tested whether the combined-movement task (Experiment 2) exacted a cost (or provided a benefit) in discriminating the probe in comparison with the single-movement conditions (Experiment 1; Fig. 2c). Discrimination performance for probes at the saccade target was approximately the same whether participants made only a saccade or made both a reach and a saccade (77% vs. 75%), $p > .05$ (independent-samples t test). Similarly, for probes at the reach target, there was no difference in discrimination performance between the single-movement and combined-movement tasks (73% vs. 72%), $p > .05$.

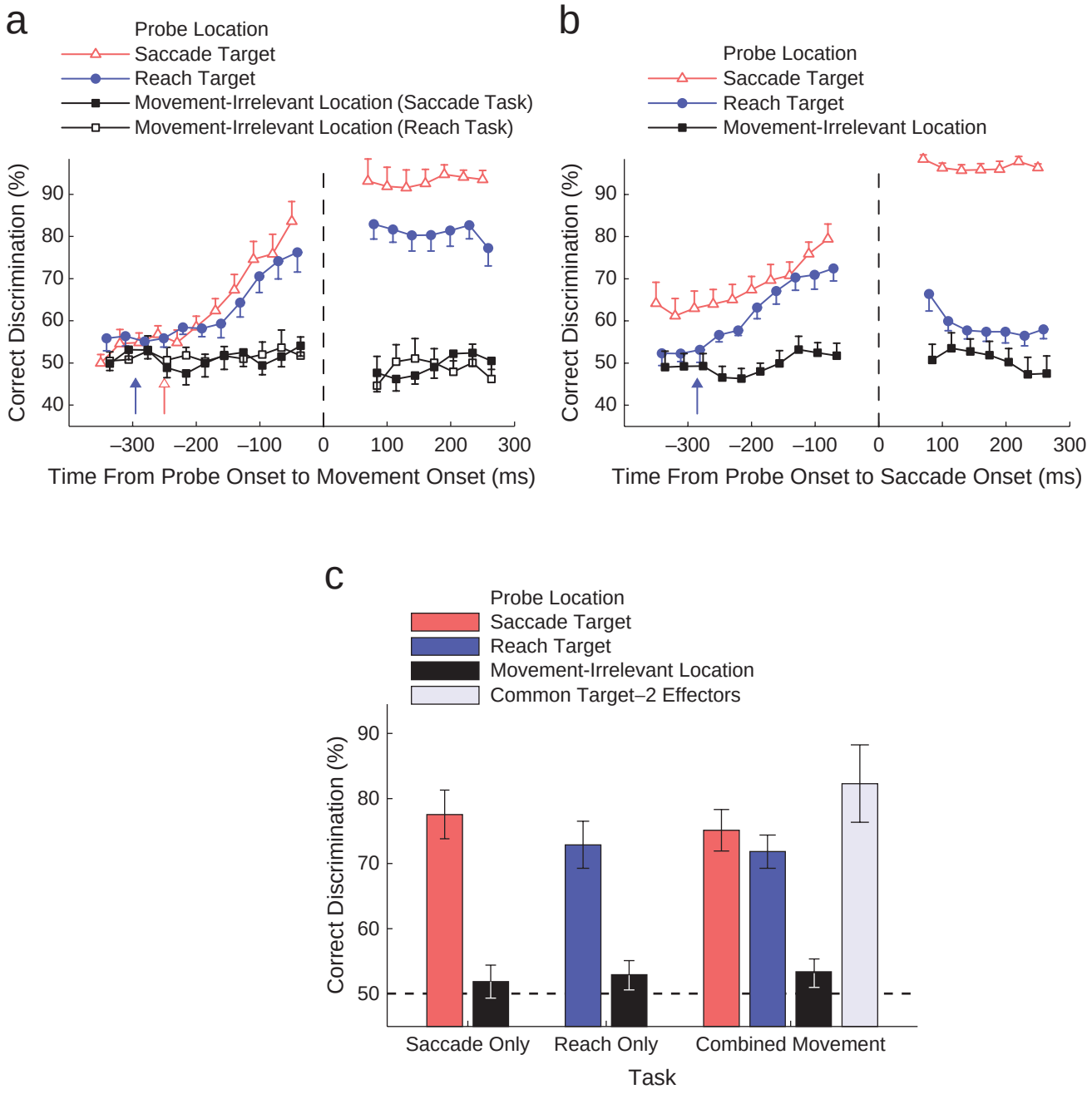


Fig. 2. Discrimination performance in Experiments 1 and 2. The graph in (a) plots the percentage of correct performance in Experiment 1 as a function of task and probe location (at the movement goal or a movement-irrelevant location) for probes appearing at various times before and after movement onset. The vertical arrows indicate the average times when the reach and saccade cues were presented, respectively. The vertical dashed line indicates the time of movement onset. The graph in (b) plots the percentage of correct performance in Experiment 2 (combined-movement task) as a function of probe location (at a movement goal or a movement-irrelevant location) for probes appearing at various times before and after saccade onset. The small vertical arrow indicates the average time when the movement cue was presented. The vertical dashed line indicates the time of saccade onset. The graph in (c) compares discrimination rates from these two experiments for probes presented less than 100 ms before movement onset. Results are shown separately for probes at the saccade target, at the reach target, at a movement-irrelevant location, and at the location of both a saccade and a reach. Error bars denote standard errors of the mean.

Thus, when simultaneous eye and hand movements were planned, there was no reduction in the attentional resources available for these two systems. This is surprising given that it has been shown previously that planning a saccade leaves few attentional resources for other, covertly attended locations (Montagnini & Castet, 2007). However, it seems that preparing a second action—with another effector system—is not subject to this fundamental limitation. We also observed that participants were better at discriminating the probes if the eye and the hand movements were directed to the same location

rather than to two different locations (Fig. 2c). This pattern was found both for the comparison between probes at a separate eye movement goal and probes at a common target (75% vs. 85%, respectively), $t(6) = -2.56$, $p = .04$, and for the comparison between probes at a separate reach goal and probes at a common target (72% vs. 85%, respectively), $t(6) = -3.44$, $p = .01$. This increase in the probe-discrimination rate shows that the processes for selecting eye and hand targets can act synergistically and indicates that separate attentional resources are used in the selection of eye and hand targets.

We next examined the degree to which the attentional selection of a saccade target and the attentional selection of a reach target are dynamically independent. We split each participant's data by median saccade latency. The short-latency saccades started on average 112 ms (*SEM* = 6 ms) earlier than the long-latency saccades, and this temporal difference was reflected in the time course of attentional allocation (Fig. 3a). It took participants 105 ms longer to reach the performance level of 75% correct probe discrimination on trials with long saccade latencies than on trials with short saccade latencies (difference determined by fitting probe discrimination for probes at the various onsets with a sigmoidal function, separately for trials with short- and long-latency saccades). In other words, attention shifted to the saccade location earlier when saccade latencies were short and later when saccade latencies were long. During the interval from 100 to 200 ms after presentation of the movement cue, participants were better at discriminating probes presented at the saccade location if saccade latencies were short than if they were long (Fig. 3a). These results demonstrate the close relationship between

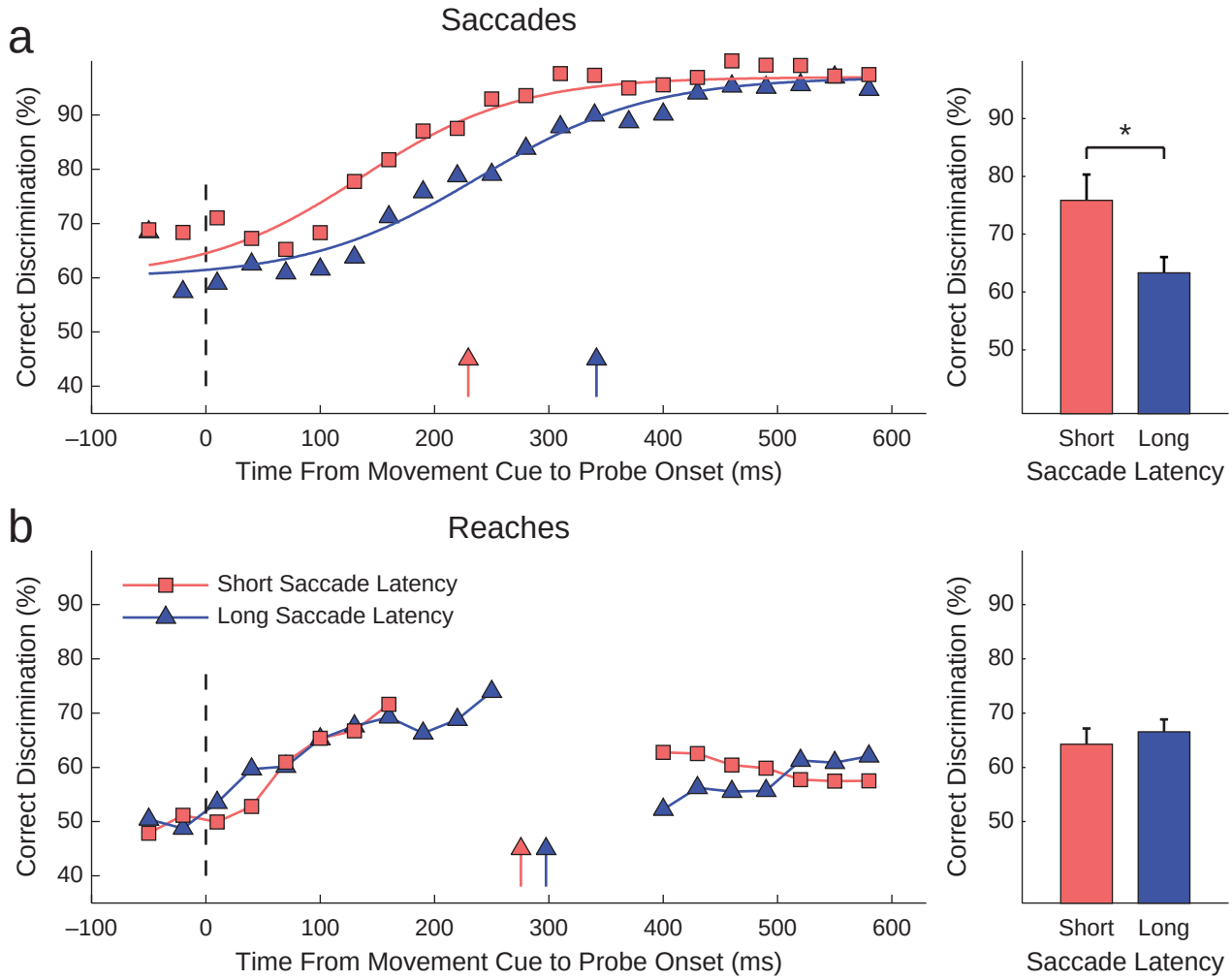


Fig. 3. Independence of attention for eye and hand movements in Experiment 2. A median split was used to categorize each participant's performance data into trials with short-latency saccades and trials with long-latency saccades. The vertical arrows denote the mean latencies of short- and long-latency saccades. The vertical dashed lines indicate the onset of the movement cue. The line graph in (a) shows probe discrimination as a function of probe onset (relative to the movement cue) and saccade latency for probes at the saccade target. The bar graph shows mean probe-discrimination performance for short- and long-latency saccades for probes at saccade targets presented 100 to 200 ms after the movement cue. The line graph in (b) shows probe discrimination as a function of probe onset and saccade latency for probes at the reach target. The bar graph shows mean probe-discrimination performance for short- and long-latency saccades for probes presented at reach targets 100 to 200 ms after the movement cue. Error bars denote standard errors of the mean. The asterisk indicates a significant paired comparison ($p < .05$).

attentional allocation and initiation of a saccade. (We found a comparable result when we analyzed the saccade-only task in Experiment 1; in contrast, we did not observe this effect for reaching movements in the reach-only task.)

In contrast, the time course of attentional allocation for probes at the reach goal was the same for trials with short- and long-latency saccades, and was thus independent from attentional allocation to the saccade location (Fig. 3b). Attention did not shift faster to the reaching location if saccade latencies were short than if they were long. In other words, no matter how early or late attention was allocated to the saccade goal, this did not affect attentional allocation at the reach goal. This finding suggests that attentional allocation to one location is dynamically independent of attentional allocation to the other.

One could object that the selective processing of the saccade and reach goals in this experiment may have resulted from a strategic allocation of attention because the probe was more likely to appear at the saccade and reach goals than at the movement-irrelevant locations. To exclude this possibility, we ran a control experiment with 7 participants. In this experiment, the probe was presented with equal probability at any of the 12 stimulus locations. The probes were shown 140 to 180 ms after the onset of the movement cue (i.e., within the last 200 ms before saccades and reaches started). We found the same pattern of results as before: If the probe was presented at one of the movement-irrelevant locations, probe discrimination was at chance level (52%). Participants performed better if the probe was presented at the saccade (68%) or reach (68%) target and best if the probe appeared at a location to which both the saccade and the reach were directed (85%). Analyses in which probe onset was matched across the experiments revealed that the results from the control experiment closely mirrored the findings of Experiment 2. In conclusion, the results from the control experiment rule out the possibility that the previous findings were due to participants strategically attending to locations where the probe was most likely to appear.

Experiment 3

It could be argued that instead of allocating attention in parallel to the two movement goals, participants may have shifted their attention to the saccade target on some trials and to the reach target on other trials. Therefore, in Experiment 3, we presented two probes at the same time in a same/different judgment task. The probes were shown for only 80 ms, a duration that is presumably too short to allow a shift of attention from one location to another. Rather, we assumed that this task could be performed successfully only if participants allocated their attention in parallel to the saccade and the reach goals.

Method

The stimulus sequence and procedure were the same as in the combined-movement task (Experiment 2) except that two probes were shown: One of them was always at one of the movement goals, and the other was either at the second movement goal (50% of trials) or at a movement-irrelevant location (50% of trials). Participants reported whether the two probes were the same or different (rather than identifying the probes). Six of the 10 observers who participated in Experiment 2 participated in this experiment. Each participant performed two blocks of 192 trials.

Results

Participants performed better than chance only when the probes were presented at the two movement goals (63%), $t(5) = 3.34$, $p = .02$. If the probability of identifying the probe at the saccade goal is p_1 and the probability of identifying the probe at the reach goal is p_2, then the probability of correctly identifying stimuli appearing at the two locations simultaneously (as in a same/different task) can be calculated as follows: $(p_1 \times p_2) + (1 - p_1) \times (1 - p_2)$. We used each participant's data from Experiment 2 to calculate his or her probability of correctly identifying the two probes in Experiment 3. Therefore, for each participant, we had the predicted probe-discrimination probability (from data in Experiment 3) and the observed probe-discrimination probability (the data in Experiment 2). The observed discrimination rate of 63% was indeed not different from the predicted discrimination rate of 62%, $p > .05$ (repeated measures t test), a result confirming that participants allocated their attention to the two locations in parallel.

Experiment 4

Experiment 4 was aimed at confirming the dynamic independence of attentional allocation to eye and hand movement targets. In this experiment, two movement cues were presented centrally, one after the other: The first cue indicated the reach target, and the second cue indicated the saccade target. The cues appeared with a stimulus onset asynchrony (SOA) of 150 or 200 ms. If attention is allocated independently to the two movement goals, the delay in attentional allocation would be expected to differ between the two SOA conditions.

Method

The stimuli and procedure were the same as in Experiment 2 except for the following. On each trial, a reach movement cue was presented for 100 ms, and participants had to reach to the indicated location. Either 50 or 100 ms after the offset of the first cue, a second movement cue appeared for 100 ms; this cue indicated the saccade location. Thus, the SOA between the first and second movement cues was either 150 or 200 ms. The distance between the saccade target and the reach target was either two or four items. Six observers participated in this experiment. Each participant performed at least four blocks of 144 trials.

Results

Attention shifted to the reach goal after the first movement cue and to the saccade target after the second movement cue (Fig. 4). There was no difference in probe-discrimination rates at the reach goal between the two SOA conditions at any of the time points before reach onset, all *p*s > .05 (repeated measures *t* test). In contrast, probe-discrimination performance at the saccade target was modulated by SOA. The probe-discrimination rate at the saccade target rose above chance level about 80 ms later for the longer-SOA condition (SOA = 200 ms) than for the shorter-SOA condition (SOA = 150 ms). Note that for the 150-ms SOA condition, probe discrimination at the saccade location was far above chance before the reach movement started (mean reach latency = 272, *SEM* = 15 ms). This means that the selection of the saccade target did not wait for the onset of the reach, but rather depended on the onset of the saccade cue. These results demonstrate the temporal independence of attentional allocation to the two movement targets and rule out the possibility that the parallel allocation of attention observed in Experiment 2 was due to the precuing of the saccade target.

General Discussion

In a series of experiments, we found that when participants made simultaneous eye and hand movements to separate locations, attention was allocated in parallel to the two locations, with no cost arising from the need to plan two movements instead of one. Therefore, even though the eye and hand systems are linked, attentional limits do not constrain selection of targets for simultaneous eye and hand movements. Furthermore, we demonstrated that delaying eye movement led to a delay in the attentional shift to the corresponding saccade target but left the attentional deployment to the reach target unaffected. This finding indicates that the attentional control mechanisms for the eye and hand are dynamically independent. Given these results, we propose that separate, effector-specific attentional controllers, instead of a unitary attentional system, are involved in distributing visual attention to multiple task-relevant locations.

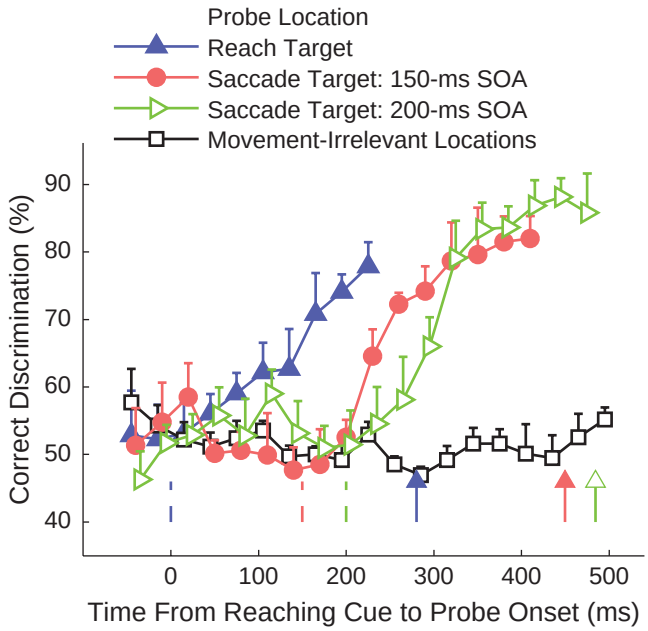


Fig. 4. Probe-discrimination performance in Experiment 4 as a function of probe onset (relative to the reaching cue), probe location, and stimulus onset asynchrony (SOA). The dashed lines denote the onsets of the movement cues: the reaching cue at 0 ms and the saccade cues at 150 and 200 ms. The vertical arrows indicate the reaching and saccade latencies. Discrimination data for the two SOA conditions are combined for probes presented at the reach target and at movement-irrelevant locations. Error bars denote standard errors of the mean.

Our experimental findings are perfectly in line with the predictions of the premotor theory of visual attention (Rizzolatti, Riggio, & Sheliga, 1994). This theory suggests the existence of multiple spatial pragmatic maps, one specific to each effector system. Neurons in these maps become activated when a movement is prepared, and attention results as a consequence of the activity of the pragmatic maps.

The alternative hypothesis is that movement goals for saccades and reaches are selected by a single, shared system representing a unitary map of action-relevant or salient objects (e.g., Itti & Koch, 2000). Indeed, the existence of such maps has been proposed for both frontal eye fields and the lateral intraparietal area, which are also implicated in the selection of saccade goals (Goldberg, Bisley, Powell, & Gottlieb, 2006; Moore, Armstrong, & Fallah, 2003). However, the assumption that these specific areas represent all salient objects is incompatible with the finding that these regions mainly represent potential saccade targets, and do not represent reach targets (Snyder et al., 1997). Also, our observation that the selection processes for eye and hand movement goals interact synergistically when the two effectors are directed to a common target is best explained by assuming that selection of eye movement goals and selection of hand movement goals occur in separate systems, rather than in a common, effector-agnostic system. Thus, both our results and current neurophysiological findings seem to indicate that the selection of movement goals is effector-specific and applies to only the objects that are relevant for the particular type of action. Interactions between the selection systems, such as the observed synergistic interaction when eye and hand movement goals were shared, may then occur through backward connections converging onto earlier visual areas (Moore et al., 2003).

Although our results have shown that the selection of eye and hand movement targets can be independent, a number of studies have found considerable cross talk between these movement systems. For example, saccade amplitudes influence reaching amplitudes (van Donkelaar, 1997), gaze is anchored to the reaching goal while people are reaching (Neggers & Bekkering, 2000), and people are likely to look where they choose to reach (Horstmann & Hoffmann, 2005). Cross-coupling has also been demonstrated in single-cell recording studies showing that eye-position signals modulate reach-related activity in parietal cortex (Batista, Buneo, Snyder, & Andersen, 1999) and that

hand-position signals modulate saccade-related activity in frontal cortex (Thura, Hadj-Bouziane, Meunier, & Boussaoud, 2008). We interpret these findings as showing that eye and hand movement systems keep track of each other, so that the eye knows where the hand will go and vice versa. These interactions may in principle occur at various stages of sensorimotor processing. However, our findings suggest that eye-hand coupling does not result from a common attentional selection mechanism, but probably follows from interactions at later processing stages.

We also demonstrated that attention can be transiently allocated to multiple locations. Whereas classical theories of attention assumed a single focus of selection, and this idea has been reinforced recently (Dubois, Hamker, & VanRullen, 2009), our data reveal that multiple foci of attention are possible when actions are planned. This idea is in line with other recent studies showing that when a sequence of eye or hand movements to multiple targets is prepared, attention spreads in parallel to all action-relevant goals, establishing spatially separate attentional foci (Baldauf & Deubel, 2010; Godijn & Theeuwes, 2003). These findings are in stark contrast to those obtained using tasks that involve intentional attention shifts: When making a saccade, people are worse at discriminating visual stimuli presented at locations other than the saccade goal (Tibber, Grant, & Morgan, 2009; Wilder, Kowler, Schnitzer, Gersch, & Dosher, 2009). Similarly, planning goal-directed pointing or simple button presses reduces performance in tasks requiring attentional shifts to other locations (Brisson & Jolicoeur, 2007; Gherri & Eimer, 2010; Wilder et al., 2009). Although attentional allocation seems to compete with movement planning in tasks involving intentional shifts of attention, attentional resources can be distributed to multiple targets without evidence of resource limitations during the planning of combined eye and hand movements, as shown here, as well as during the preparation of movement sequences. This suggests a dissociation between attentional shifts that occur for the purpose of action preparation and those that are involved in purely perceptual tasks.

In conclusion, we have demonstrated that selective attention is allocated in parallel to the targets of eye and hand movements, and we propose that the attentional control mechanisms for these two effector systems are largely independent. This finding highlights the flexibility of the visuomotor system in being able to simultaneously select and process multiple objects relevant for different actions and suggests that signals from separate sources are related to target selection for different effectors.

Declaration of Conflicting Interests

The authors declared that they had no conflicts of interest with respect to their authorship or the publication of this article.

Funding

This research was supported by the Deutsche Forschungsgemeinschaft (GRK 1091) and the Seventh Framework Program of the European Community (Project “GRASP,” ICT–215821).

References

Baldauf, D., & Deubel, H. (2010). Attentional landscapes in reaching and grasping. *Vision Research*, *50*, 999–1013.

Batista, A.P., Buneo, C.H., Snyder, L.H., & Andersen, R.A. (1999). Reach plans in eye-centered coordinates. *Science*, *285*, 257–260.

Bekkering, H., Adam, J.J., van den Aarssen, A., Kingma, H., & Whiting, H.T. (1995). Interference between saccadic eye and goal-directed hand movements. *Experimental Brain Research*, *106*, 475–484.

Beurze, S.M., de Lange, F.P., Toni, I., & Medendorp, W.P. (2009). Spatial and effector processing in the human parietofrontal network for reaches and saccades. *Journal of Neurophysiology*, *101*, 3053–3062.

Brisson, B., & Jolicoeur, P. (2007). Electrophysiological evidence of central interference in the control of visuospatial attention. *Psychonomic Bulletin & Review*, *14*, 126–132.

Calton, J.L., Dickinson, A.R., & Snyder, L.H. (2002). Non-spatial, motor-specific activation in posterior parietal cortex. *Nature Neuroscience*, *5*, 580–588.

Curtis, C.E., & Connolly, J.D. (2008). Saccade preparation signals in the human frontal and parietal cortices. *Journal of Neurophysiology*, *99*, 133–145.

Deubel, H., & Schneider, W.X. (1996). Saccade target selection and object recognition: Evidence for a common attentional mechanism. *Vision Research*, *36*, 1827–1837.

Deubel, H., & Schneider, W.X. (2003). Delayed saccades, but not delayed manual aiming movements, require visual attention shifts. *Annals of the New York Academy of Sciences*, *1004*, 289–296.

Deubel, H., Schneider, W.X., & Paprotta, I. (1998). Selective dorsal and ventral processing: Evidence for a common attentional mechanism in reaching and perception. *Visual Cognition*, *5*, 81–107.

Dickinson, A.R., Calton, J.L., & Snyder, L.H. (2003). Nonspatial saccade-specific activation in area LIP of monkey parietal cortex. *Journal of Neurophysiology*, *90*, 2460–2464.

Dubois, J., Hamker, F.H., & VanRullen, R. (2009). Attentional selection of noncontiguous locations: The spotlight is only transiently “split.” *Journal of Vision*, *9*(5), Article 3. Retrieved from http://www.journalofvision.org/content/9/5/3

Gherri, E., & Eimer, M. (2010). Manual response preparation disrupts spatial attention: An electrophysiological investigation of links between action and attention. *Neuropsychology*, *48*, 961–969.

Godijn, R., & Theeuwes, J. (2003). Parallel allocation of attention prior to the execution of saccade sequences. *Journal of Experimental Psychology: Human Perception and Performance*, *29*, 882–896.

Goldberg, M.E., Bisley, J.W., Powell, K.D., & Gottlieb, J. (2006). Saccades, salience and attention: The role of the lateral intraparietal area in visual behavior. *Progress in Brain Research*, *155*, 157–175.

Horstmann, A., & Hoffmann, K.P. (2005). Target selection in eye-hand coordination: Do we reach to where we look or do we look to where we reach? *Experimental Brain Research*, *167*, 187–195.

Itti, L., & Koch, C. (2000). A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision Research*, *40*, 1489–1506.

Johansson, R.S., Westling, G., Backstrom, A., & Flanagan, J.R. (2001). Eye-hand coordination in object manipulation. *The Journal of Neuroscience*, *21*, 6917–6932.

Kowler, E., Anderson, E., Dosher, B., & Blaser, E. (1995). The role of attention in the programming of saccades. *Vision Research*, *35*, 1897–1916.

Land, M.F., Mennie, N., & Rusted, J. (1999) The roles of vision and eye movements in the control of activities of daily living. *Perception*, *28*, 1311–1328.

Lawrence, B.M., & Snyder, L.H. (2009). The responses of visual neurons in the frontal eye field are biased for saccades. *The Journal of Neuroscience*, *29*, 13815–13822.

Levy, I., Schluppeck, D., Heeger, D.J., & Glimcher, P.W. (2007). Specificity of human cortical areas for reaches and saccades. *The Journal of Neuroscience*, *27*, 4687–4696.

Linnell, K.J., Humphreys, G.W., McIntyre, D.B., Laitinen, S., & Wing, A.M. (2005). Action modulates object-based selection. *Vision Research*, *45*, 2268–2286.

Montagnini, A., & Castet, E. (2007). Spatiotemporal dynamics of visual attention during saccade preparation: Independence and coupling between attention and movement planning. *Journal of Vision*, *7*(14), Article 8. Retrieved from http://www.journalofvision.org/content/7/14/8

Moore, T., Armstrong, K.M., & Fallah, M. (2003). Visuomotor origins of covert spatial attention. *Neuron*, *40*, 671–683.

Neggers, S.W.F., & Bekkering, H. (2000). Ocular gaze is anchored to the target of an ongoing pointing movement. *Journal of Neurophysiology*, *83*, 639–651.

Pelz, J., Hayhoe, M.M., & Loeber, R. (2001). The coordination of eye, head, and hand movements in a natural task. *Experimental Brain Research*, *139*, 266–277.

Prablanc, C., Echallier, J.F., Komilis, E., & Jeannerod, M. (1979). Optimal response of eye and hand motor systems in pointing at a visual target. *Biological Cybernetics*, *35*, 113–124.

Rizzolatti, G., Riggio, L., & Sheliga, B.M. (1994). Space and selective attention. In C. Umilta & M. Moscovitch (Eds.), *Attention and performance XV: Conscious and nonconscious information processing* (pp. 231–265). Cambridge, MA: MIT Press.

Sailer, U., Eggert, T., Ditterich, J., & Straube, A. (2000). Spatial and temporal aspects of eye-hand coordination across different tasks. *Experimental Brain Research*, *134*, 163–173.

Snyder, L.H., Batista, A.P., & Andersen, R.A. (1997). Coding of intention in the posterior parietal cortex. *Nature*, *386*, 167–170.

Song, J.-H., & McPeek, R.M. (2009). Eye-hand coordination during target selection in a pop-out visual search. *Journal of Neurophysiology*, *102*, 2681–2692.

Thompson, A.A., & Westwood, D.A. (2007). The hand knows something that the eye does not: Reaching movements resist the Müller-Lyer illusion whether or not the target is foveated. *Neuroscience Letters*, *426*, 111–116.

Thura, D., Hadj-Bouziane, F., Meunier, M., & Boussaoud, D. (2008). Hand position modulates saccadic activity in the frontal eye field. *Behavioural Brain Research*, *186*, 148–53.

Tibber, M.S., Grant, S., & Morgan, M.J. (2009). Oculomotor responses and visuospatial perceptual judgments compete for common limited resources. *Journal of Vision*, *9*(12), Article 21. Retrieved from http://www.journalofvision.org/content/9/12/21

Tosoni, A., Galati, G., Romani, G.L., & Corbetta, M. (2008). Sensory-motor mechanisms in human parietal cortex underlie arbitrary visual decisions. *Nature Neuroscience*, *11*, 1446–1453.

Van Der Werf, J., Jensen, J., Fries, P., & Medendorp, W.P. (2010). Neuronal synchronization in human posterior parietal cortex during reach planning. *The Journal of Neuroscience*, *30*, 1401–1412.

van Donkelaar, P. (1997). Eye–hand interactions during goal-directed pointing movements. *NeuroReport*, *8*, 2139–2142.

Wilder, J.D., Kowler, E., Schnitzer, B.S., Gersch, T.M., & Dosher, B.A. (2009). Attention during active visual tasks: Counting, pointing, or simply looking. *Vision Research*, *49*, 1017–1031.

Accurate grasping requires attentional resources.

Constanze Hesse[1,2] & Heiner Deubel[2]

23/11/2010

RUNNING HEAD: "visual attention in grasping"

keywords: grasping, visual attention, motor control, dual–task

words: 7235 *figures: 5*

Correspondence should be addressed to:

Dr. Constanze Hesse

Cognitive Neuroscience Research Unit

Wolfson-Research-Institute

Durham University / Queen's Campus

University Boulevard

Stockton on Tees

TS17 6BH

United Kingdom

e-mail: constanze.hesse@durham.ac.uk

phone: ++44 (0)191 33 404 47

[1]Cognitive Neuroscience Research Unit, Durham University, UK
[2]Allgemeine und Experimentelle Psychologie, Department Psychologie, Ludwig-Maximilians-Universität, München, Germany

Abstract

We investigated the effects of visuo–spatial attention on movement kinematics by employing a dual–task paradigm. Participants had to grasp cylindrical objects of different sizes (motor task) while simultaneously identifying a target digit presented at a different spatial location within a rapid serial visual presentation (perceptual task). The grasping kinematics in this dual–task situation were compared with those measured in a single task condition. Likewise, the identification performance was also measured in a single–task condition. Additionally, we kept the visual input constant across conditions by asking participants to fixate. Without instructions about the priority of tasks (Experiment 1) participants showed a considerable drop of identification performance (perceptual task) in the dual–task condition. Regarding grasping kinematics, the concurrent perceptual task resulted in a less accurate adaptation of the grip to object size in the early phase of the movement, while movement times and maximum grip aperture were unaffected. When participants were instructed to focus on the perceptual task (Experiment 2), the identification performance stayed at about the same level in the dual–task and the single–task conditions. The perceptual improvement was however associated with a further decrease in the accuracy of the early grip adjustment. We conclude that visual attention is needed for the effective control of the grasp kinematics, especially for a precise adjustment of the hand to object size when approaching the object.

Introduction

Before initiating a goal–directed grasping movement, the target object has to be selected from the visual scene. Visual attention is the mechanism which underlies this kind of selective processing. In short, visual attention fulfills two important functions: On the one hand, attention supports perception by facilitating the detection of certain stimuli (Posner, 1980), and on the other hand, visual attention is involved in the selection of objects that are relevant for goal–directed actions, thereby helping to specify the spatial parameters of a movement (Neumann, 1987; Allport, 1987).

It has been proposed that both mechanisms ("selection for perception" and "selection for action") share a common attentional resource (Schneider, 1995). So far, this theory is mainly supported by the finding that selectional processes preparing a spatio–motor action bind the attentional mechanisms in visual perception to the movement target. For example, while preparing a pointing or grasping movement, visual discrimination performance is increased at the selected movement positions, whereas the discrimination performance is reduced close to chance level at positions which are not associated with an upcoming movement (e.g., Deubel, Schneider, & Paprotta, 1998; Deubel & Schneider, 2004; Schiegg, Deubel, & Schneider, 2003; Baldauf, Wolf, & Deubel, 2006). Thus, the sensorimotor system seems to selectively allocate attention to relevant movement–related positions in space when planning a movement. Note that attention can be distributed between several objects of interest in parallel, for instance when obstacles have to be taken into account (Deubel & Schneider, 2004), or movements are performed bimanually (Baldauf & Deubel, 2008). Although there are many studies showing that visual attention is deployed to the goal positions of the movement well in advance leaving only little processing capacity for action-irrelevant items in the visual field, there are considerably less studies looking for the complementary effects of an attentional task on movement kinematics.

Many everyday activities involve simultaneous cognitive tasks and motor control activities, and can obviously be well performed by healthy humans (e.g., grasping a coffee mug while talking on the phone). On the one hand, it could be argued that some motor tasks, such as eye–movements or grasping, are immune to interference since they are assumed to occur "automatically", thus not requiring central cognitive resources (e.g., Shiffrin and Schneider, 1977; 1984; for an overview, see Norman and Shallice, 2000). On the other hand, assuming that the attentional capacity available is limited (Broadbent, 1958; 1982), performing two tasks at the same time could be expected to result in interferences. These inconsistent predictions on the occurrence of interferences between attentional and motor tasks are also reflected in the research examining the effects of dividing attention on smooth pursuit eye–movements. Whereas some researchers observed impairments in the accuracy of smooth pursuit eye–movements when an attentionally demanding secondary task had to be performed (Chen, Holzman, & Nakayama, 2002; Hutton & Tegally, 2005), other researchers reported an even enhanced pursuit performance when employing a dual task paradigm (van Gelder, Lebedev, Liu, & Tsui, 1995; Kathmann, Hochrein, & Uwer, 1999). The latter, rather counterintuitive finding was explained by proposing that pursuit eye tracking is a highly automatic process that is performed best in the absence of controlled attention (Kathmann et al., 1999).

In contrast to the extensive research done on interference effects between eye movements and visual attention, studies investigating attention–related effects on pointing and grasping movements have primarily focused on the problem of whether and how the presence of a distractor in the workspace object modifies the movement kinematics (e.g., Bonfiglioli & Castiello, 1998; Castiello, 1996; Kritikos, Bennett, Dunai, & Castiello, 2000; Tipper, Lortie, & Baylis, 1992; Tipper, Howard, & Jackson, 1997; Jackson, Jackson, & Rosicky, 1995). The findings of these studies suggest that distractor objects only interfere with movements when they become task–relevant (therefore attracting more attention) and share similar properties with the target object. For example, in the study of Castiello (1996), participants had to count

how often a distractor object was illuminated while executing a grasping movement (covert attention). When the distractor object was a large object, maximum grip aperture was larger than when the distractor object was a small object, although the size of the target object remained constant. Thus, it was concluded that task–irrelevant properties of the distractor are automatically processed activating in parallel a motor program for the distractor object which in turn causes the observed interference effects. In short, this shows that when attention has to be divided between a distractor and a target object, the grasp parameterization is influenced by the distractor's properties. The assumption that grasping requires attentional resources is further supported by recent studies conducted in our lab showing that the introduction of a secondary (motor) task can lead to sequencing effects in grasp pre–shaping (Hesse & Deubel, 2010). In a free-viewing condition, grip aperture was not adapted to the size of the target object unless a concurrently executed pointing movement (performed with the other hand) was finished. However, when fixation was required, both tasks (grasping and pointing) could be well performed in parallel.

In all the studies discussed so far, the secondary task was always another motor task, and when distractors were used they were related to the grasping movement. In this study, we applied a dual–task paradigm in order to test whether a pure visual task requiring attentional resources interferes with grasp programming and execution. Therefore, we asked participants to simultaneously perform a grasping movement to a target object while trying to detect a target digit in a rapid serial visual presentation of digits presented at a different spatial location. In order to avoid the effects of overtly changing attention between the perceptual and the motor tasks, participants were asked to keep fixation when performing both tasks. We were especially interested in the question of whether grasp kinematics were altered when a simultaneous perceptual task (requiring attention) had to be performed. We additionally examined the complementary effects of the grasping movement on the visual identification performance. Finally, the perceptual and motor performance reached in the dual–task conditions (grasping

and identifying) were compared to the performance reached in matched perceptual and visuo-motor single–task conditions, respectively.

Experiment 1

Methods

Participants

Twelve undergraduate and graduate students of the Ludwig–Maximilians–University Munich (five men; mean age = 28, age range: 21–47) participated in the experiment. They were paid 8 Euro per hour of participation. All participants were right–handed by self report, had normal or corrected–to–normal visual acuity, and were naive with respect to the purpose of the study. The experiments were done with the understanding and written consent of each participant and conformed to the Declaration of Helsinki.

Apparatus and stimuli

Three black wooden rings served as target objects. All rings had an inner annulus of 25 mm, but differed in their outer diameters (diameters 50, 55, and 60 mm).

Participants sat comfortably on an adjustable chair within a dimly lit room. They looked straight at a transparent Plexiglas pane (34 cm x 30 cm x 0.5 cm) which was placed vertically on the tabletop at a viewing distance of 50 cm (see Figure 1A). A chin rest was used to maintain a constant head position throughout the experiment. In every trial two rings of different size were attached to the Plexiglas pane. The rings were vertically aligned with a distance of 8.5 cm between their centers (see Figure 1B). At a distance of 100 cm behind the pane a video projector was installed projecting onto the back of the Plexiglas to which a transparent foil and a light gray paperboard were attached. Three holes were cut in the paperboard allowing the projector to

project at the position of the inner annuli and the position of fixation. The fixation location was placed centrally between the rings and 5.5 cm to their left to prevent interference with grasping movements that were performed with the right hand. The starting position of the hand was marked by a pin which was affixed on the table top. The distance between starting pin and target ring was 38 cm for the lower and 42 cm for the upper target position.

The projector was used to present the fixation cross and the attentional (visual) stimuli in the annuli of both rings. The visual stimuli consisted of a rapid serial visual presentation (RSVP) of digits (between 1 and 9). The digits were white projected on a gray background for 50 ms with a blank interval of 75 ms between each presentation. The size of the digits was $2.7^{o}$of visual angle. The size and the presentation duration of the digits in the RSVP were determined in a pilot study adjusting the digits such that participants achieved on average an identification performance of approximately 85%.

Trajectories of the grasping movements were recorded using a Polhemus Liberty electromagnetic motion tracking system at a sampling rate of 240 Hz. The Polhemus Liberty tracking system provides 6-degrees-of-freedom (position and orientation) information at a static accuracy of 0.8 mm RMS for the x, y and z positions and 0.15 deg for sensor orientation. The Polhemus sensors were attached to the nails of the thumb and the index finger of the right hand (using adhesive pastels: UHU-patafix, UHU GmbH, Bühl, Germany and medical tape). Prior to the experiment a calibration procedure was used to align the Cartesian coordinate system (x,y,z) of the Polhemus system such that the start position on the table corresponded with the point of origin (0,0,0). Also, the orientation signals of the sensors attached to index finger and thumb were calibrated to a standard orientation. By considering the individual thickness of index finger and thumb, the orientation information allowed us to calculate the grasp touch points of thumb and index finger relative to the sensors, for each sample recorded during the experiment. During the experiment participants wore liquid–crystal shutter glasses

(Milgram, 1987), which rapidly suppress vision by changing from a transparent to an opaque state.

Procedure

Participants began each trial with the index finger and thumb of the dominant right hand located at the starting position. Before the beginning of each trial the shutter glasses turned opaque and the experimenter arranged the objects on the Plexiglas pane. After the experimenter had placed both rings, he/she initiated the trial manually by pressing a key. When the shutter glasses became transparent participants looked at the fixation cross located to the left of the objects. Simultaneously the presentation of the RSVP in both annuli began. After the fixation period which lasted for 1s, the fixation cross turned into an arrow cuing either the upper or the lower annulus. Depending on the block, the cue indicated to the participants at which target location they had to detect the target digit and/or to which target they had to direct their grasping movement, respectively. There were three different task blocks: 1) *grasping baseline:* In this block the cue indicated to the participants which ring they had to grasp. The RSVPs could be ignored and no target digit was presented. 2) *perception baseline:* In this block the cue indicated to the participants to which annulus they had to direct their attention. Black target digits were presented in both annuli and participants had to report the digit that was presented in the cued annulus. No grasping movements were required in these trials. 3) *dual–task condition:* In this block, participants had to do both, grasping the target ring while simultaneously directing the attention to the opposite annulus reporting the black target digit presented within the RSVP. The cue indicated to the participants to which annulus they had to direct their attention. In the perceptual baseline and dual–task conditions the target digit (which had to be identified by the participants) appeared randomly 200 ms, 350 ms or 500 ms after the cue presentation (that signalled the beginning of the movement). We chose different presentation times in order to prevent participants from predicting the occurrence of the target digit during the experiment.

Furthermore, we aimed at presenting the target during the time the movement was initiated since the movement programming phase is supposed to be most crucial for the distribution of attentional capacities (Schiegg et al., 2003; Baldauf & Deubel, 2010). The mean RT associated with cued prehension is approximately 450 ms according to Jakobson and Goodale (1991). The RSVP was restricted such that the two digits occurring simultaneously in both ring locations were never identical in one pass. In all blocks participants were instructed to keep fixation at cue location for the whole duration of the trial. After three seconds, the shutter glasses turned opaque and the experimenter returned the objects and prepared the next trial.

Insert Figure 1 about here

In the trials which required grasping a target ring, participants grasped the ring with index finger and thumb (precision grip), and then put the object in front of them on the tabletop. When participants had to report the target digit, they did so in the end of each trial. The reported digit was then entered by the experimenter sitting next to the participant. If participants did not perceive the target they were instructed to guess. Furthermore, they were instructed to start their movements immediately after the cue was presented and to do both tasks in the dual–task block as accurately as possible.

There were six different combinations of ring sizes (see Figure 1C) and two possible target positions (up and down). In each trial the combinations of target rings and cued location were determined pseudo–randomly. In the baseline trials each combination was presented two times resulting in 24 trials. Thus, in the grasping baseline each target size was actually grasped 8 times. In the perception baseline the target letter was presented 8 times in each ring size and the three presentation times were assigned randomly to the 24 trials (each presentation time occurring eight times but independent of the combination of ring sizes). In the dual–task trials each combination of the six ring sizes and the two cued locations (“up” vs. “down”) was

presented five times resulting in 60 trials (i.e. each ring size was grasped 20 times). Again the presentation times of the target letter were assigned randomly with each delay occurring 20 times during the 60 trials, and each presentation time occurring at least 5 times for each object size.

Before starting each block, six practice trials were executed for familiarization with the task. The sequence of blocks was counterbalanced across participants. Before the experiment started the position of the digits in the rings was individually adjusted such that participants perceived the digits as presented in the middle of the annuli.

Data Processing

The visual identification performance and the kinematics of the grasping movements measured in the dual–task condition were compared with the performance in the baseline conditions respectively. The percentage of correctly identified target digits was used as indicator for the perceptual performance and compared between the dual–task and the perceptual baseline condition. Furthermore, we determined how often, in trials in which the target digit was reported erroneously, the reported digit corresponded to the digit which presented opposite to the cued location to which the grasping movement was directed.

In order to determine the effects of the perceptual task on grasping movements we compared certain kinematic parameters between the dual–task condition and the grasping baseline condition. The finger trajectories were filtered off-line using a second-order Butterworth filter that employed a low-pass cut–off frequency of 15 Hz. Movement velocities were determined by differentiating the position signal of the markers. Movement onset was defined by a velocity criterion. The first frame in which the wrist exceeded a velocity threshold of 0.1 m/s was taken as movement onset. Reaction time (RT) was defined as the time between the cue presentation and movement onset. The first frame in which the velocity of the wrist dropped below a threshold of 0.1m/s was taken as the touch of the object. Movement time (MT) was defined as

the time between movement onset and touch of the object. Furthermore, we determined the approach to the target location by measuring the trajectory of the fingers, calculated as the virtual midpoint between index finger and thumb, along the y-axis and z-axis (see Figure 1 for axis assignments). The trajectory data was determined every 20 ms from movement onset. Additionally, several parameters related to the grasp component of the movement were quantified. Maximum grip aperture (MGA) was defined as the maximum distance in 3D between the calculated grasp positions of the thumb and the index finger during MT. Moreover, the time when MGA was reached was determined. Finally, in order to determine how well the aperture was adjusted to the size of the object over time we first computed the size of the aperture as mean values binned over 10 samples (42 ms) from movement onset. Then we conducted a linear regression analysis in order to determine the slope of the function relating object size to aperture size over time. This provided a sensitive measure of the adjustment of grip aperture to the specific objects sizes during the grasp.

Since we were mainly interested in the effects of object size on grasp kinematics in the baseline conditions (grasping only) and in the dual–task conditions (grasping and simultaneous perceptual task), the grasping data was averaged over the two ring positions (up and down) and the different ring combinations. Furthermore, we checked in a pre–analysis for the effects of presentation time of the target digit on grasping kinematics and perceptual performance (see the sections on the pre–processing of the data). Since the presentation time was found to show no major effects on our dependent variables, the data was averaged over all presentation times, and then further analysed using repeated measures analysis of variance (3x2 ANOVA) with the factors ring size (50 mm, 55 mm, 60 mm) and task (baseline condition vs. dual–task condition). A significance level of α=0.05 was used for the statistical analyses. If the sphericity assumption was violated, the degrees of freedom were adjusted using the Greenhouse–Geisser correction (Greenhouse & Geisser, 1959). Values are presented as means ± standard errors of the mean.

Results

Perception

Pre–analysis on the effects of target presentation time

In order to test for the effects of the different presentation times of the target letter on the perceptual performance, we applied a repeated–measures ANOVA with the factor presentation time (200 ms, 350 ms, 500 ms) to the data collected in the perceptual baseline and in the dual–task conditions. The perceptual performance in the baseline conditions was unaffected by the presentation time, $F(2,22)=0.9$, $p=.79$. In the dual–task conditions, there was a slight tendency for a better identification performance when the target was presented later (58.3% ± 4.1% for presentation after 200 ms, 64.3% ± 4.0% after 350 ms, and 67.1% ± 3.5% after 500 ms). However, the finding failed to reach significance, $F(2,22)=2.7$, $p=.09$. In the following analyses we merged the data over all presentation times.

Baseline vs. Dual–task condition

Regarding the identification performance in the visual attention task we were interested in how the additional grasping task affected the performance compared to the baseline condition in which no concurrent movement was required. The identification performance was averaged over all ring combinations, ring sizes, and presentation times in both conditions. On average, participants identified 84.4% ± 3.4% of the digits correctly in the baseline condition. This performance dropped significantly in the dual–task conditions, $t(11)=6.8$, $p<.001$, in which participants only identified 63.1% ± 3.2% of the target digits correctly (see Figure 2). When we examined the erroneous trials more closely, it turned out that participants reported the digit which was presented in the opposite annulus significantly more often in the dual–task conditions than in the baseline conditions, $t(11)=3.3$, $p=.007$. In the baseline conditions the

opposite digit was reported in 12.0% ± 5.1% of the erroneous trials which corresponded approximately to the chance level (11.1%). In contrast, in the dual–task conditions the opposite digit was reported in 26.3% ± 4.3% of all erroneous trials. This data is in line with previous findings showing that movements directed to a certain location in space bind attentional resources, resulting in a reduced ability to allocate attention to other positions in space (Deubel & Schneider, 2004; Baldauf et al., 2006; Baldauf & Deubel, 2010).

Insert Figure 2 about here

Grasping

Pre–analysis on the effects of target presentation time

To test whether the time of target presentation affected grasping kinematics, we applied a repeated–measures ANOVA with the factor presentation time (200 ms, 350 ms, 500 ms) to the data collected in the dual–task condition. No significant effect of presentation time was observed on any of the variables of interest: MGA: $F(2,22)=1.4$, $p=.28$, time to MGA: $F(2,22)=1.1$, $p=.36$, MT: $F(2,22)=1.3$, $p=.29$, and RT: $F(2,22)=1.5$, $p=.25$. For the following analyses we merged the data over all presentation times.

Baseline vs. Dual–task condition

Transport

Figure 3a shows the mean movement paths of the hand (calculated as the virtual mid–point between index finger and thumb) in y–direction and two–dimensionally in y–z space (from the start position to the target location) averaged over the different ring sizes and ring positions. Surprisingly, the trajectory in the baseline and the dual–task trials are virtually identical. Thus, superficially there seems to be no indication that the approach to the target object was affected by the simultaneously performed attention task. This conclusion is supported by the MT data.

The MTs were neither affected by the size of the object to grasped, $F(2,22)=0.2$, $p=.79$ nor by the task, $F(1,11)=2.0$, $p=.19$. There was no interaction effect ($p=.91$). It took participants on average 614*ms* ± 13*ms* in the baseline condition to perform the movement, and 594*ms* ± 16*ms* in the dual–task conditions. Thus, contrary to our expectations, MTs were not prolonged when an additional attention task had to be performed. Regarding the RTs, we found a tendency for prolonged movement initiation times in the dual–task compared to the baseline task, $F(1,11)=4.8$, $p=.05$. On average, participants initiated their movement after 385*ms* ± 21*ms* in the baseline conditions and after 429*ms* ± 17*ms* in the dual–task conditions, while there was no effect of object size and no interaction (both $p>.48$). This result is in line with the finding that doing two tasks simultaneously results in dual–task costs, typically reflected in an increase in error rates and reaction times as compared to doing only one task at a time (Pashler, 1994; Schubert, 2008).

Insert Figure 3 about here

Pre–shaping

In a second step we questioned whether the perceptual task affects the grasp pre–shaping. We had hypothesized that an attentional task may prevent the early perceptual processing of the grasp target, such that the movement–relevant parameters of the object, i.e. its size, could not be integrated during the early movement phase. A very reliable and commonly used measure to quantify the adjustment of the grip to object size is MGA (Smeets & Brenner, 1999). As expected, a 3 (object size) x 2 (task: baseline vs. dual–task) repeated–measures ANOVA revealed a significant effect of object size, $F(2,22)=81.6$, $p<.001$. On average the size of MGA was 65.7*mm* ± 1.5*mm* for the small object, 70.1*mm* ± 1.7*mm* for the medium sized object and 73.3*mm* ± 1.6*mm* for the large object. However, we observed no significant main effect of task, $F(1,11)=0.05$, $p=.82$ and no interaction effect ($p=.53$). This finding indicates that the MGA was equally well adapted to object size in both conditions suggesting no effect of the perceptual task

on grip scaling. Regarding the timing of MGA we found, however, a small but significant effect of task, $F(1,11)=8.2$, $p=.02$. On average MGA was reached after $482ms \pm 22ms$ in the baseline conditions and after $525ms \pm 21ms$ in the dual–task conditions. There was no effect of object size and no interaction (both $p>.36$). Thus, although the MGA was about the same size for the different objects in the baseline and in the dual–tasks, it was reached a bit later when a perceptual task had to be performed simultaneously. This finding prompted us to look more closely at the adjustment of the grip over time.

For this purpose, we calculated the size of the aperture in time–bins of 10 samples (42 ms). Figure 4A shows the aperture profiles for the different object sizes in the baseline and the dual–task conditions. In both conditions the aperture shows a smooth opening over time. A closer look at the figure reveals that the fingers open a bit slower in the dual–task conditions and that the aperture profiles seem to separate later for the different object sizes. To examine this observation in more detail, we calculated the slope of the function relating object size to aperture size using linear regression analysis. This measure reflects the integration of object size in the grip adjustment over time. Firstly, we tested again for the effects of presentation time of the target letter on grip scaling in the dual–task conditions. Therefore, we applied a repeated–measures ANOVA with the factors presentation time (200 ms, 350 ms, 500 ms) and time bin to the data collected in the dual–task condition. Again there was no significant effect of presentation time ($p=.80$) and no significant interaction between presentation time and time bin ($p=.25$). As expected, the main effect of time bin was highly significant, $F(19,209)=17.9$, $p<.001$. On basis of these findings, we averaged the data in the dual–task conditions over all presentation times.

Figure 5 shows the average grip scaling over time in the baseline condition and in the dual–task conditions. The slopes increased much slower in the dual–task condition which required reporting the target digit presented within the RSVP, than in the baseline conditions in which no perceptual task was performed. The repeated–measures ANOVA with the factors task

and time-bin revealed a significant interaction effect, $F(19,209)=2.1$, $p=.007$, suggesting that in both tasks the slopes changed differently over time. As expected there was a significant effect of time, $F(19,209)=61.4$, $p<.001$, reflecting the increase of the slopes over the course of the movement. The main effect of condition failed to reach the level of significance, $F(1,11)=4.6$, $p=.06$. However, we would not have assumed that the slopes between the baseline and the dual–task conditions vary per se but that the slopes increase later and/or slower in the dual–task condition compared to the baseline condition as confirmed by the interaction effect. When calculating the differences between conditions at each time point using paired–samples t-tests, four comparisons became significant.

Insert Figure 4 about here

Insert Figure 5 about here

Experiment 2

The results of Experiment 1 show that whereas the perceptual performance suffers considerably when doing a simultaneous motor task, the effects of the attention task on the motor performance are more subtle. Surprisingly, neither movement times nor the trajectories changed when the perceptual task had to be performed. The only indication that the perceptual task interfered with the motor planning was found in the adjustment of the grip aperture to object size. One reason why grasping kinematics remained relatively unaffected by the secondary task might have been that participants prioritized performing the grasping task over the perceptual task, since the consequences of failing in the motor task were more relevant (e.g. dropping the object). If the decrease in motor performance is due to the imposed cognitive demands, increasing the level of difficulty of the perceptual task should result in a further decrease of the grasping performance. Thus, we conducted a second experiment in which we made the perceptual task more difficult and additionally instructed participants to try to keep

their recognition performance in the dual task as good as in the baseline condition (i.e. to set priority to the perceptual task).

Methods

Participants

The same twelve participants as in Experiment 1 participated in this experiment. Again, all participants were naive with respect to the purpose of the study.

Stimuli and Procedure

The apparatus and the stimuli were identical to those used in Experiment 1. We only varied the difficulty of the perceptual task by increasing the speed of the RSVP and decreasing the size of the digits. The digits were again presented for 50 ms but a shorter blank interval of 55 ms between each presentation was used. The size of the numbers now was 2.1 degrees of visual angle. In addition we varied the instruction given to the participants: When doing the dual–task block participants were asked to keep their identification performance as good as possible. As in Experiment 1, the dual–task block consisted of 60 trials. Moreover, we measured the perceptual baseline in which participants were asked to report the target number presented in the previously cued target annulus without performing a grasping movement. The data was analyzed identically to Experiment 1. The grasping kinematics observed in the dual–task were compared to the grasping baseline measured in Experiment 1 using a 2 (task) x 3 (object size) repeated–measures ANOVA. The order of blocks was counterbalanced across participants.

Results

Perception

Pre–analysis on the effects of target presentation time

As in Experiment 1, we tested for the effects of the different presentation times of the target digit on the perceptual performance in the perceptual baseline and in the dual–task conditions. For this purpose, we applied a repeated–measures ANOVA with the factor presentation time (200 ms, 350 ms, 500 ms) to the data. In both conditions (baseline and dual–task) the perceptual performance was unaffected by the time of target presentation (both $p>.35$). Thus, we merged the data of all presentation times for further analyses.

Baseline vs. Dual–task condition

Regarding the identification performance in the visual attention task, we were interested in whether our instruction to maintain a good identification performance reduced the performance differences between the baseline and the dual–task conditions as observed in Experiment 1. The identification performance was averaged over all ring combinations and ring sizes in both conditions. On average, participants correctly identified 73.0% ± 3.2% of the digits in the baseline condition (the drop of recognition performance compared to Experiment 1 reflects the increased difficulty of the task). Amazingly, this performance stayed at about the same level in the dual–task conditions, $t(11)=0.18$, $p=.86$, in which participants identified 72.4% ± 3.5% of the target digits correctly (see Figure 2). This result demonstrates that the participants were well able to set different priorities to the perceptual task if asked to do so. As in Experiment 1 there was an increased probability to report the digit which was presented in the opposite annulus in the dual–task conditions (22.2% ± 2.9%) as compared to the perceptual baseline conditions (14.4% ± 4.0%). In contrast to Experiment 1 this trend did not become significant, $t(11)=1.9$, $p=.08$.

Grasping

Pre–analysis on the effects of target presentation time

As in Experiment 1, we tested for the effects of target presentation time on grasping kinematics in the dual–task condition by applying a repeated–measures ANOVA with the factor presentation time (200 ms, 350 ms, 500 ms) to the data. No significant effect of presentation time was observed for time to MGA, MT, and RT (all $p>.55$). There was, however, a significant effect of presentation time on the size of MGA, $F(2,22)=5.2$, $p=.02$. Post–hoc comparisons showed that the size of MGA was smaller when the target occurred after 500 ms than when the target was presented after 200 ms. For further analyses we merged the data of all presentation times.

Baseline vs. Dual–task condition

Transport

As in Experiment 1, the movement times were unaffected by performing the perceptual task, even when its difficulty was increased. On average, movements took $638ms \pm 22ms$ which was not significantly different from the MTs observed in the baseline conditions of Experiment 1, $F(1,11)=0.72$, $p=.42$. Again there was a marginal effect of the perceptual task on RTs when comparing them to the RTs of the baseline condition of Experiment 1, $F(1,11)=3.8$, $p=.07$. On average, participants initiated their movements after $454ms \pm 25ms$. Again, RTs and MTs were unaffected by the size of the object (all $p>.37$).

Pre–shaping

Regarding the size of MGA, we found no significant difference between the, now more difficult, dual–task condition and the baseline condition as measured in Experiment 1, $F(1,11)=1.7$, $p=.22$. As expected, the repeated–measures ANOVA with the factor ring size (small, medium, large) showed that the size of MGA was significantly affected by object size, $F(2,22)=67.2$, $p<.001$. On average the size of MGA was $69.4mm \pm 3.1mm$ for the small object, $73.4mm \pm 3.3mm$ for the medium sized object, and $76.4mm \pm 3.1mm$ for the large object. On

average the MGA was reached after 557*ms* ± 26*ms* in this experiment. Unlike in Experiment 1 this value did not differ significantly from the baseline condition, $F(1,11)=3.7$, $p=.08$. There was no effect of object size on the timing of MGA ($p=.37$).

As shown in Experiment 1, the more meaningful parameter than the size and timing of MGA was however the adaptation of the grip to the object size over time. Again, we checked first for the effects of presentation time of the target letter on grip scaling in the dual–task conditions. For this purpose, we applied a repeated–measures ANOVA with the factors presentation time (200 ms, 350 ms, 500 ms) and time bin to the data collected in the dual–task condition. As in Experiment 1, there was no significant main effect of presentation time ($p=.45$) and no interaction between presentation time and time bin ($p=.40$). The main effect of time bin was highly significant, $F(19,209)=40.8$, $p<.001$, however. For further analyses, the data was averaged over all presentation times in the dual–task conditions.

Figure 4B depicts the averaged aperture profiles when grasping objects of different sizes. In comparison to the findings of Experiment 1 the aperture profiles separate even later in this experiment (visual inspection of the figure reveals that during the first 350 ms the aperture opening is virtually identical for all object sizes). This observation is further supported by the calculation of the slopes of the function relating grip aperture to object size. Figure 5 shows that the grip adjustment was indeed further impaired by making the perceptual task more difficult and asking participants to prioritize this task over grasping. The repeated–measures ANOVA revealed again a significant interaction effect between time and task, $F(19,209)=3.5$, $p<.001$. Moreover, the main effects of time, $F(19,209)=80.5$, $p<.001$, and task $F(1,11)=7.5$, $p=.02$ were significant, thus indicating that the slopes increased over time but were significantly lower than in the baseline condition. Post–hoc tests indicated that all differences between the fourth (147 ms) and eleventh (441 ms) time bin were significantly lower than in the baseline conditions.

Discussion

It has repeatedly been shown that visual attention is allocated to the target positions of reaching and grasping movements when preparing an action, suggesting a coupling between selection for action and selection for perception in these tasks (Deubel et al., 1998; Schiegg et al., 2003; Baldauf & Deubel, 2008). The purpose of this study was to examine whether there is also an inverse effect of withdrawing visual attention from a grasping task on movement kinematics.

The main finding across both experiments was that a demanding secondary task requiring visual attention led to an impairment of the early adjustment of grip aperture to object size. Interestingly, the effect of the perceptual task on grasping kinematics was limited to the manipulation component of the movement. Neither the movement trajectory nor movement times - both measures related to the transport component of the movement - changed when participants were asked to perform a simultaneous identification task. This finding could be related to the proposition that the transport and the manipulation components of a grasping movement are controlled by two independent, though temporally coupled, visuo-motor channels (Jeannerod, 1981; 1984). Studies investigating the effects of paying (covert) attention to distractor objects reported that interference effects only occurred when target and distractor involved the programming of different parameters for the same grasping component. For example, Castiello (1996) found that the size of a distractor object that had to be attended covertly selectively influenced the size of the grip aperture when grasping a target object (for similar results see also Kritikos et al., 2000). Complementary, when covert attention had to be paid to a moving distractor, interference effects were observed in the transport component only (Bonfiglioli & Castiello, 1998). However, to our knowledge no study has yet shown that even a purely visual task, being of no direct relevance for the reach–to–grasp movement, influences the accuracy of movement programming and execution. One possible reason why we observed a selective impairment in the adjustment of the manipulation component in our study might be

that we varied the size of the target object from trial to trial, whereas the objects were presented at constant locations ("up" or "down"). It is possible that participants quickly learned the trajectories towards these locations and automatized the transport component of the movement. Automatic movement control is performed without controlled attention and is thus less susceptible to interference processes. Moreover, there is evidence from anatomical and lesion studies in humans and monkeys that the transport and the manipulation components are controlled by different neural structures of the brain (e.g., Jeannerod, Arbib, Rizzolatti, & Sakata, 1995; Taira, Mine, Georgopoulos, Murata, & Sakata, 1990; Castiello, 2005).

Furthermore, the effects on grasping kinematics were limited to the early phase of the grip adaptation. The adaptation of the MGA to object size (which occurs in the second half of the movement between 60% and 75% of movement time; see, Jeannerod, 1981; 1984; Smeets and Brenner, 1999) was largely unaffected by the secondary perceptual task. This finding is possibly a direct consequence of the dual–task paradigm since the target digit was always presented at the beginning of the movement (at the latest 500 ms after cue–presentation). Assuming that it took participants approximately 400 ms to initiate the movement, most of the visuo–perceptual processing was done during the movement initiation phase and shortly after. Thus, computational resources had to be shared between the tasks during movement preparation. Close to the end of the movement the target digit was already identified, and resources were freed and could fully be used to perform the grasping task. The finding also suggests that movement programming takes place during the movement initiation phase as withdrawing attention at this time results in a higher inaccuracy in the specification of some kinematic parameters.

A second interesting finding of this study was that a concurrent grasping movement resulted in a significant drop of performance in the perceptual task (compared to the single task condition). In other words, when no instructions were given regarding the priority of the tasks (Experiment 1), we observed a strong decrement in the perceptual performance whereas the

changes in grasping kinematics were less conspicuous. Thus, participants seemed to prioritize the visuo-motor over the perceptual task, if not instructed otherwise. Similar findings have been reported in dual–task paradigms investigating the relation between cognitive tasks and walking performance (e.g., Li, Lindenberger, Freund, & Baltes, 2001), although effects in these studies were primarily confined to elderly people. However, compared to walking and postural control, grasping is a fine motor skill and therefore possibly more easily disturbed by a secondary task. One potential reason why participants try to keep their performance up in the grasping task might be that inaccuracies in this task have direct negative consequences, such as dropping or breaking the object. In comparison, reporting a wrong number in the perceptual task is not associated with any immediate consequence for the participant.

Besides, in the dual–task conditions, participants tended to report the target presented at the grasping location more frequently than chance level would predict. This finding gives additional evidence that during grasping some attention is automatically deployed to the position of the grasp, facilitating the visuo–spatial discrimination performance at this location (Schiegg et al., 2003; Baldauf & Deubel, 2010). Here, we were able to demonstrate that this effect is accompanied by a withdrawal of attention from positions that are not related to the grasp, even occurring when these grasp–unrelated positions would actually require attention in order to perform a secondary task successfully. This finding is in line with the propositions of the Visual Attention Model (VAM) of Schneider (1995) suggesting that "selection–for–action" and "selection–for–perception" are performed by a common visual attention mechanism. That is, visual recognition of one target is assumed to delay the motor selection of another target, and vice versa.

Finally, we showed in the second experiment that participants were able to keep their identification performance in the dual–task condition as good as in the single–task condition when they were instructed to focus on the perceptual task. This result gives further evidence that humans can flexibly shift attention between tasks depending on instructions (Kelly, Janke,

& Shumway-Cook, 2010). However, the enhanced perceptual performance was only achieved at the expense of an additional accuracy impairment regarding the early grip adaptation to object size.

Taken together, our findings show that there are dual task costs when a grasping movement and a perceptual task that requires visual attention are performed simultaneously, indicating that both tasks compete for limited computational resources (Broadbent, 1958; 1982; Baldauf & Deubel, 2010). Hence, grasping seems to be a process which requires some attentional capacities, challenging the proposition that such movements are performed completely automatized. The allocation of attention to action–irrelevant items in the visual field leads to a poorer adaptation of the grasp to the object's properties which might partly explain why humans tend to drop objects more often when they are distracted.

Acknowledgement

This study was supported by the 7th Framework Program of the European Community (project "GRASP", ICT-215821) and by the Cluster of Excellence "Cognition for Technical Systems" (Project 301). Constanze Hesse currently holds a postdoctoral research fellowship of the German Research Council (DFG/HE 6011/1-1).

References

Allport, D. A. (1987). Selection for action: some behavioral and neurophysiological considerations of attention and action. In H. Heuer & A. F. Sanders (Eds.), *Perspectives on perception and action* (pp. 395–419). Hillsdale, NJ: Lawrence Erlbaum Associates Inc.

Baldauf, D., & Deubel, H. (2008). Visual attention during the preparation of bimanual movements. *Vision Research*, *48*, 549–563.

Baldauf, D., & Deubel, H. (2010). Attentional landscapes in reaching and grasping. *Vision Research*, *50*(11), 999–1013.

Baldauf, D., Wolf, M., & Deubel, H. (2006). Deployment of visual attention before sequences of goal–directed hand movements. *Vision Research*, *46*, 4355–4374.

Bonfiglioli, C., & Castiello, U. (1998). Dissociation of covert and overt spatial attention during prehension movements: Selective interference effects. *Perception and Psychophysics*, *60*(8), 1426–1440.

Broadbent, D. E. (1958). *Perception and communication*. New York: Oxford University Press.

Broadbent, D. E. (1982). Task combination and selective intake of information. *Acta Psychologica*, *50*(3), 253–290.

Castiello, U. (1996). Grasping a fruit: Selection for action. *Journal of Experimental Psychology: Human Perception and Performance*, *22*(3), 582–603.

Castiello, U. (2005). The neuroscience of grasping. *Nature Reviews Neuroscience*, *6*(10).

Chen, Y., Holzman, P. S., & Nakayama, K. (2002). Visual and cognitive control of attention in smooth pursuit. *Progress in Brain Research*, *140*(140), 255–265.

Deubel, H., & Schneider, W. X. (2004). Attentional Selection in sequential manual movements, movements around an obstacle and in grasping. In G. W. Humphreys & M. J. Riddoch (Eds.), *Attention in Action* (pp. 69–91). Hove: Psychological Press.

Deubel, H., Schneider, W. X., & Paprotta, I. (1998). Selective dorsal and ventral processing: evidence for a common attentional mechanism in reaching and perception. *Visual Cognition*, *5*, 81–107.

Greenhouse, S. W., & Geisser, S. (1959). On methods in the analysis of profile data. *Psychometrika*, *24*(2), 95–112.

Hesse, C., & Deubel, H. (2010). Bimanual movement control is moderated by fixation strategies. *Experimental Brain Research*, *202*(4), 837–850.

Hutton, S. B., & Tegally, D. (2005). The effects of dividing attention on smooth pursuit eye tracking. *Experimental Brain Research*, *163*(3), 306–313.

Jackson, S. R., Jackson, G. M., & Rosicky, J. (1995). Are non–relevant objects represented in working–memory? The effect of nontarget objects on reach and grasp kinematics. *Experimental Brain Research*, *102*, 519–530.

Jakobson, L. S., & Goodale, M. A. (1991). Factors affecting higher–order movement planning: A kinematic analysis of human prehension. *Experimental Brain Research*, *86*, 199–208.

Jeannerod, M. (1981). Intersegmental coordination during reaching at natural visual objects. In J. Long & A. Baddeley (Eds.), *Attention and Performance* (Vol. 9, pp. 153–168). Hillsdale, NJ: Erlbaum.

Jeannerod, M. (1984). The timing of natural prehension movements. *Journal of Motor Behavior*, *16*(3), 235–254.

Jeannerod, M., Arbib, M. A., Rizzolatti, G., & Sakata, H. (1995). Grasping objects: The cortical mechanisms of visuomotor transformation. *Trends in Neurosciences*, *18*, 314–320.

Kathmann, N., Hochrein, A., & Uwer, R. (1999). Effects of dual task demands on the accuracy of smooth pursuit eye movements. *Psychophysiology*, *36*(2), 158–163.

Kelly, V. E., Janke, A. A., & Shumway-Cook, A. (2010). Effects of instructed focus and task difficulty on concurrent walking and cognitive task performance in healthy young adults. *Exp Brain Res*, *207*(1-2), 65–73.

Kritikos, A., Bennett, K. M. B., Dunai, J., & Castiello, U. (2000). Interference from distractors in reach–to–grasp movements. *The Quarterly Journal of Experimental Psychology*, *53A*(1), 131–151.

Li, K. Z., Lindenberger, U., Freund, A. M., & Baltes, P. B. (2001). Walking while memorizing: age–related differences in compensatory behavior. *Psychological Science*, *12*(3), 230–237.

Milgram, P. (1987). A spectacle–mounted liquid–crystal tachistoscope. *Behavior Research Methods, Instruments, & Computers*, *19*(5), 449–456.

Neumann, O. (1987). Beyond capacity: a functional view of attention. In H. Heuer & A. F. Sanders (Eds.), *Perspectives on perception and action* (pp. 361–394). Hillsdale, NJ: Lawrence Erlbaum Associates Inc.

Norman, D. A., & Shallice, T. (2000). Attention to Action: Willed and Automatic Control of Behavior. In M. S. Gazzaniga (Ed.), *Cognitive neuroscience: A reader* (pp. 376–390). Massachusetts, USA: Blackwell Publishers Inc.

Pashler, H. E. (1994). Dual–task interference in simple tasks: data and theory. *Psychological Bulletin*, *116*, 220–244.

Posner, M. I. (1980). Orienting of attention. *Quarterly Journal of Experimental Psychology*, *32*(1), 3–25.

Schiegg, A., Deubel, H., & Schneider, W. X. (2003). Attentional selection during preparation of prehension movements. *Visual Cognition*, *10*(4), 409–431.

Schneider, W. X. (1995). VAM: A neuro–cognitive model for visual attention control of segmentation, object recognition ans space–based motor action. *Visual Cognition*, *2*, 331–375.

Schubert, T. (2008). The central attentional limitation and executive control. *Frontiers in Bioscience*, *13*, 3569–3580.

Shiffrin, R. M., & Schneider, W. (1977). Controlled and automatic human information processing: II. perceptual learning, automatic attending and a general theory. *Psycholological Review*, *84*(2), 127–190.

Shiffrin, R. M., & Schneider, W. (1984). Automatic and controlled processing revisited. *Psychological Review*, *91*(2), 269–276.

Smeets, J. B. J., & Brenner, E. (1999). A new view on grasping. *Motor Control*, *3*, 237–271.

Taira, M., Mine, S., Georgopoulos, A. P., Murata, A., & Sakata, H. (1990). Parietal cortex neurons of the monkey related to the visual guidance of hand movement. *Exp Brain Res*, *83*(1), 29–36.

Tipper, S. P., Howard, L. A., & Jackson, S. R. (1997). Selective reaching to grasp: Evidence for distractor interference effects. *Visual Cognition*, *4*, 1–38.

Tipper, S. P., Lortie, C., & Baylis, G. C. (1992). Selective reaching: Evidence for action–centered attention. *Journal of Experimental Psychology: Human Perception and Performance*, *18*, 891–905.

van Gelder, P., Lebedev, S., Liu, P. M., & Tsui, W. H. (1995). Anticipatory saccades in smooth pursuit: task effects and pursuit vector after saccades. *Vision Research*, *35*, 667–678.

Figure Legends

1. A: Schematic drawing of the experimental set–up used (not drawn to scale). B: Arrangement of the stimuli on the Plexiglas pane (front view) C: All six possible ring combinations. Each ring-combination could be cued in either "up" or "down" indicating to the participant which object to grasp and/or which location to attend in the perceptual task.
2. Identification performance in the perceptual baseline conditions and in the dual–task conditions of Experiment 1 (no instructions as to task priority), and in Experiment 2 (instruction to set priority to the perceptual task). Chance level was 11.1%. Error bars depict ± 1 SEM between subjects.
3. A: Averaged movement trajectories in y-direction for the dual–task and baseline conditions as a function of time. B: Averaged movement path in y–z–direction for the dual–task and baseline conditions plotted separately for the upper (solid lines) and the lower (dashed lines) target position.
4. A: Experiment 1: Averaged aperture profiles when grasping objects of different sizes in the baseline condition (solid lines) and in the dual–task conditions (dashed lines) as a function of time. B: Experiment 2: Averaged aperture profiles when grasping objects of different sizes in the dual–task condition. In this experiment participants were instructed to focus on the perceptual task.
5. Adjustment of the grip scaling (i.e. slope of the function relating grip aperture to object size) in the baseline condition and in the dual–task conditions of Experiment 1 and 2, plotted as a function of time.

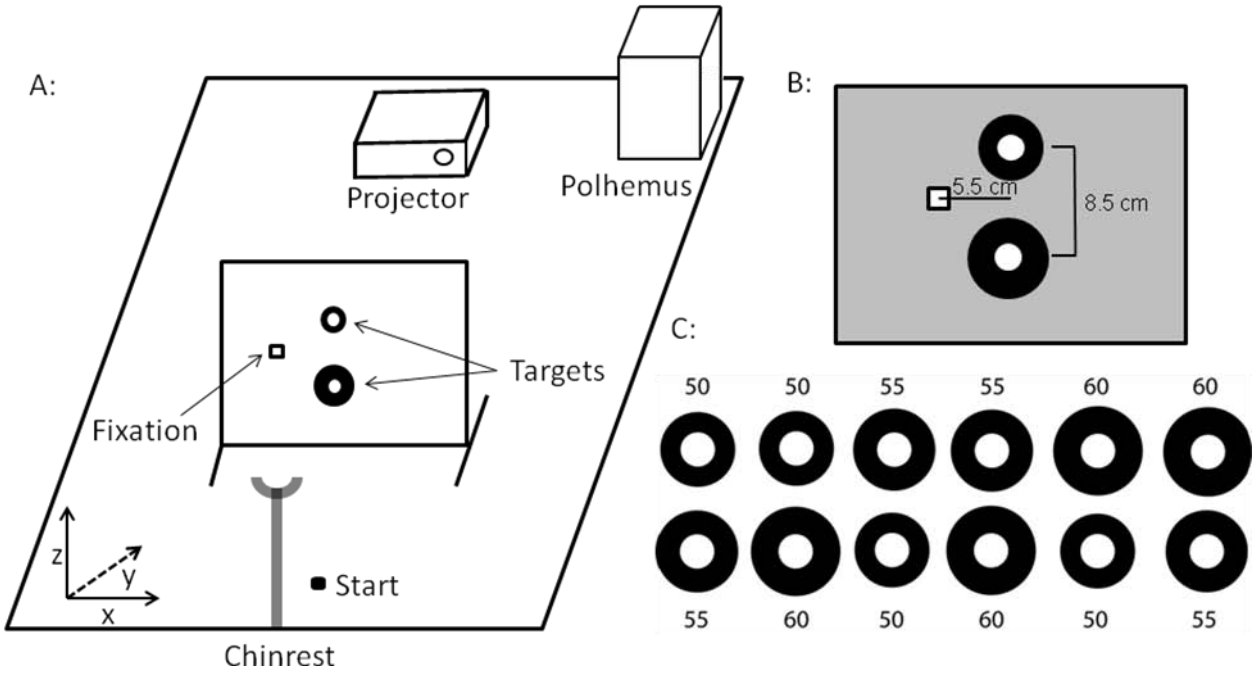

A:
Projector
Polhemus
Targets
Fixation
z
y
x
Start
Chinrest
B:
5.5 cm
8.5 cm
C:
50 50 55 55 60 60
55 60 50 60 50 55

Figure 1

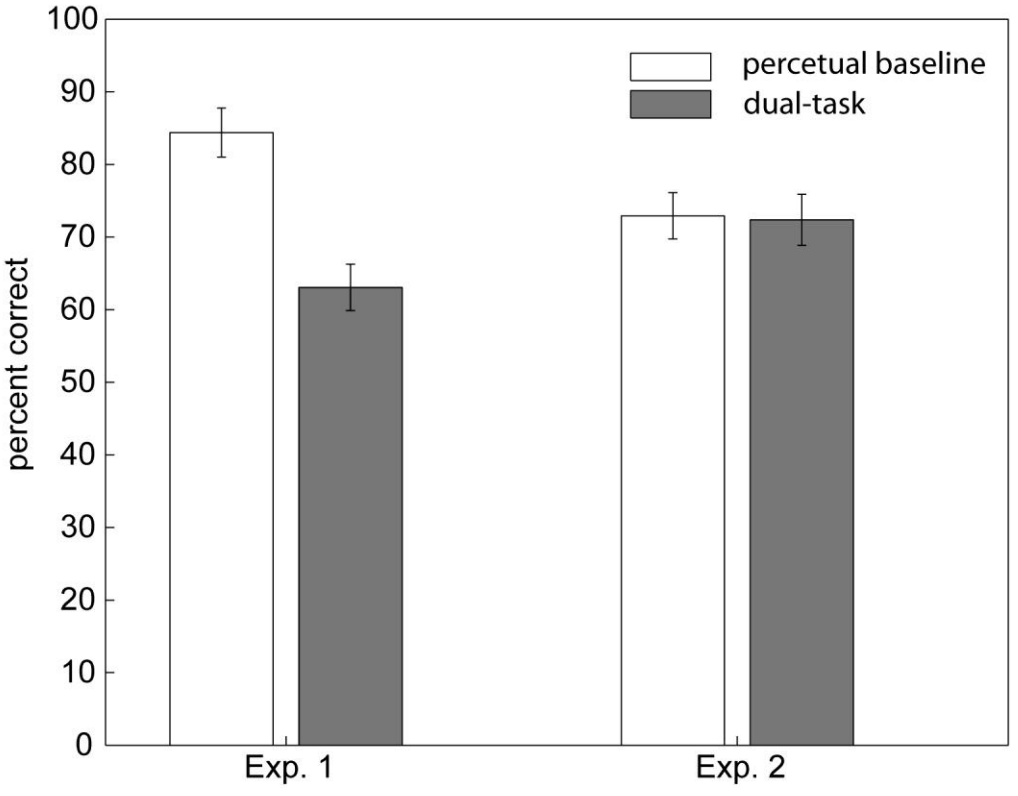

100
90
80
70
60
50
40
30
20
10
0
percent correct
percetual baseline
dual-task
Exp. 1
Exp. 2

Figure 2

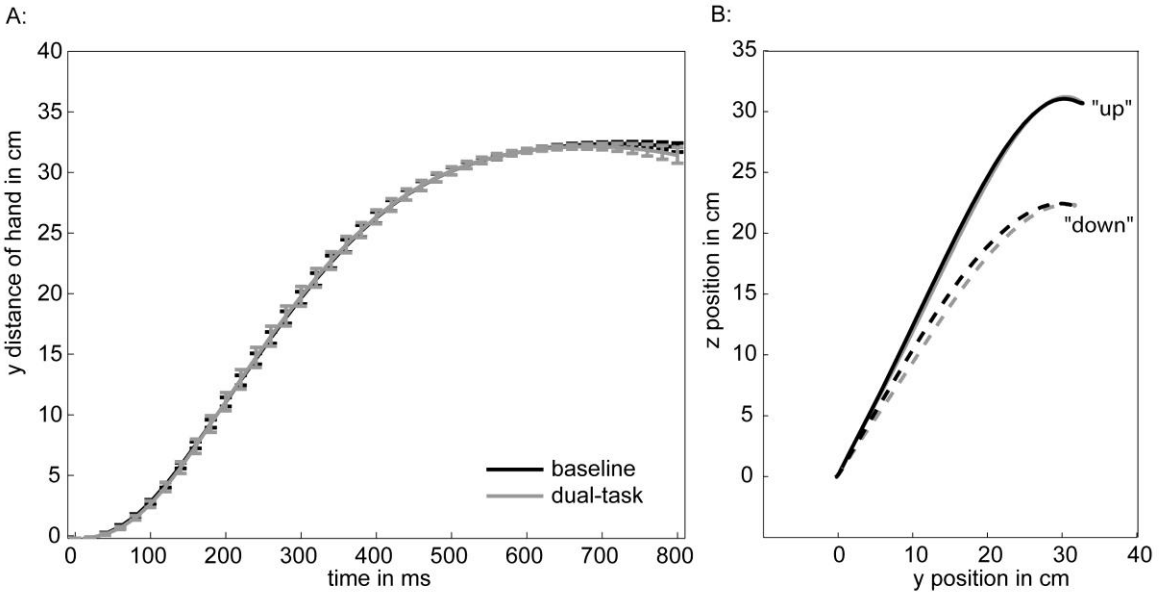

A:
B:
y distance of hand in cm
time in ms
baseline
dual-task
z position in cm
y position in cm
"up"
"down"

Figure 3

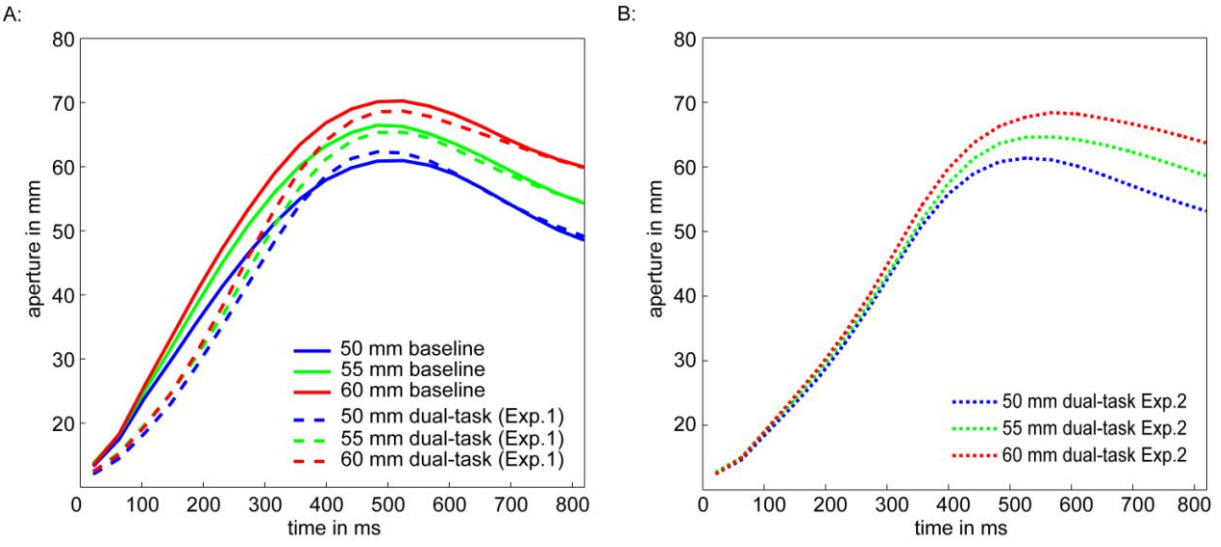

A:
B:
aperture in mm
time in ms
50 mm baseline
55 mm baseline
60 mm baseline
50 mm dual-task (Exp.1)
55 mm dual-task (Exp.1)
60 mm dual-task (Exp.1)
50 mm dual-task Exp.2
55 mm dual-task Exp.2
60 mm dual-task Exp.2

Figure 4

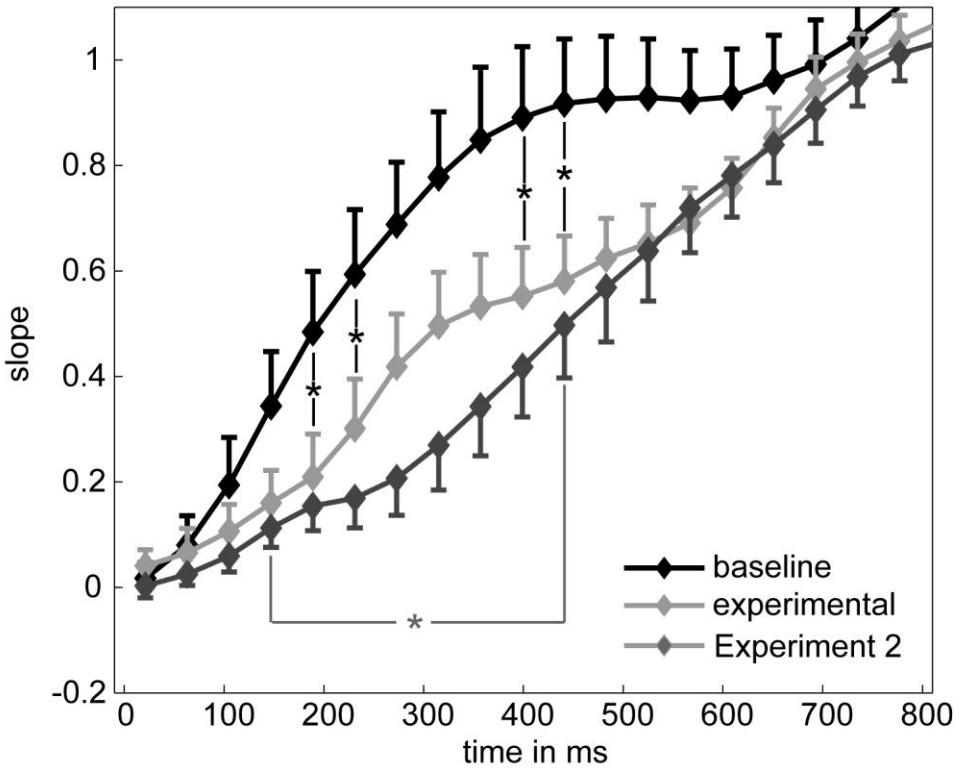

slope
time in ms
baseline
experimental
Experiment 2
1
0.8
0.6
0.4
0.2
0
-0.2
0
100
200
300
400
500
600
700
800
*
*
*
*
*

Figure 5

Markerless and Efficient 26-DOF Hand Pose Recovery

Iasonas Oikonomidis, Nikolaos Kyriazis, and Antonis A. Argyros

Institute of Computer Science, FORTH
and
Computer Science Department, University of Crete
{oikonom,kyriazis,argyros}@ics.forth.gr
http://www.ics.forth.gr/cvrl/

Abstract. We present a novel method that, given a sequence of synchronized views of a human hand, recovers its 3D position, orientation and full articulation parameters. The adopted hand model is based on properly selected and assembled 3D geometric primitives. Hypothesized configurations/poses of the hand model are projected to different camera views and image features such as edge maps and hand silhouettes are computed. An objective function is then used to quantify the discrepancy between the predicted and the actual, observed features. The recovery of the 3D hand pose amounts to estimating the parameters that minimize this objective function which is performed using Particle Swarm Optimization. All the basic components of the method (feature extraction, objective function evaluation, optimization process) are inherently parallel. Thus, a GPU-based implementation achieves a speedup of two orders of magnitude over the case of CPU processing. Extensive experimental results demonstrate qualitatively and quantitatively that accurate 3D pose recovery of a hand can be achieved robustly at a rate that greatly outperforms the current state of the art.

1 Introduction

The problem of effectively recovering the pose (3D position and orientation) of human body parts observed by one or more cameras is interesting because of its theoretical importance and its diverse applications. The human visual system exhibits a remarkable ability to infer the 3D body configurations of other humans. A wide range of applications such as human-computer interfaces, etc, can be built provided that this fundamental problem is robustly and efficiently solved [1]. Impressive motion capture systems that employ visual markers [2] or other specialized hardware have been developed. However, there is intense interest in developing markereless computer-vision based solutions, because they are non-invasive and, hopefully, cheaper than solutions based on other technologies (e.g., electromagnetic tracking).

The particular problem of 3D hand pose estimation is of special interest because by understanding the configuration of human hands we are in a position

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 744–757, 2011.

to build systems that may interpret human activities and understand important aspects of the interaction of a human with her/his physical and social environment. Despite the significant amount of work in the field, the problem remains open and presents several theoretical and practical challenges due to a number of cascading issues. Fundamentally, the kinematics of the human hand is complicated. Complicated kinematics is hard to accurately represent and recover and also yields a search space of high dimensionality. Extended self-occlusions further complicate the problem by generating incomplete and/or ambiguous observations.

1.1 Related Work

A significant amount of literature has been devoted to the problem of pose recovery of articulated objects using visual input. Moeslund et al [1] provide a thorough review covering the general problem of visual human motion capture and analysis. The problems of recovering the pose of the human body and the human hand present similarities such as the tree-like connectivity and the size variability of the articulated parts. However, a human hand usually has consistent appearance statistics (skin color), whereas the appearance of humans is much more diverse because of clothing.

A variety of methods have been proposed to capture human hand motion. Erol et al [3] present a review of such methods. Based on the completeness of the output, they differentiate between partial and full pose estimation methods, further dividing the last class into appearance-based and model-based ones.

Appearance-based methods estimate hand configurations from images directly after having learnt the mapping from the image feature space to the hand configuration space [4, 5, 6, 7]. The mapping is highly nonlinear due to the variation of hand appearances under different views. Further difficulties are posed by the requirement for collecting large training data sets and the accuracy of pose estimation. On the positive side, appearance based methods are usually fast, require only a single camera and have been successfully employed for gesture recognition.

Model-based approaches employ a 2D or 3D hand model [8, 9, 10, 11]. In the case of 3D hand models the hand pose is estimated by matching the projection of the model to the observed image features. The task is then formulated as a search problem in a high dimensional configuration space, which induces a high computational cost. Important issues to be addressed by such methods include the efficient construction of realistic 3D hand models, the dimensionality reduction of the configuration space and the development of techniques for fast and reliable hand posture estimation.

This paper presents a novel, generative method that treats the 3D hand pose recovery problem as an optimization problem that is solved through Particle Swarm Optimization (PSO). Under the taxonomy of [3], the present work can be categorized as a full, model-based pose estimation method that employs a single hypothesis. The method may integrate observations from an arbitrary number of available views without requiring special markers. This is clearly demonstrated by our decision to consider all free problem parameters jointly and simultaneously.

As a direct consequence, contrary to the work of [10], our formulation of the problem allows for a clear and effortless treatment of self-occlusions. PSO has been already applied for human pose recovery in [12], however this is done in a hierarchical fashion in contrast to our joint optimization approach. Additionally, the method of [12] is not directly applicable to hand pose recovery because stronger occlusions must be handled given weaker observation cues.

Being generative, the approach explores an essentially infinite configuration space. Thus, the accuracy of estimated pose is not limited by the size and content of the employed database, as e.g. in [7]. To the best of our knowledge, this is the first work that demonstrates that PSO can be applied to the problem of 3D hand pose recovery and solve it accurately and robustly. This is demonstrated in sequences with highly complex hand articulation where the hand is observed from relatively distant views. Additionally, it is demonstrated that the careful selection of inherently data parallel method components permits the efficient, near real-time 3D hand pose estimation and gives rise to the fastest existing method for model-based hand pose recovery.

The rest of this paper is organized as follows. Section 2 describes in detail the proposed method. Section 3 presents results from an extensive quantitative and qualitative experimental evaluation of the proposed method. Finally, Sec. 4 summarizes the paper by drawing the most important conclusions of this work.

2 Methodology

The proposed method can be summarized as follows. Observations of a human hand are acquired from a static, pre-calibrated camera network. For each observation, skin color detection and edge detection are performed to extract reference features. A 3D model of a human hand is adopted that consists of a collection of parameterized geometric primitives. Hand poses are represented by a total of 27 parameters that redundantly encode the 26 degrees of freedom of the human hand. Given the hand model, poses which would reproduce the observations are hypothesized. For each of them, the corresponding skin and edge feature maps are generated and compared against their reference counterparts. The discrepancy between a given pose and the actual observation is quantified by an error function which is minimized through Particle Swarm Optimization (PSO). The pose for which this error function is minimal constitutes the output of the proposed method at a given moment in time. Temporal continuity in hand motion is assumed. Thus, the initial hypotheses for current time instance are restricted in the vicinity of the solution for the previous time instant. The method incorporates computationally expensive processes which cannot be adequately handled by conventional CPU processing. However, the exploitation of the inherent data parallelism of all the required components through a GPU powered implementation, results in near real-time computational performance. The following sections describe in more detail the components outlined above.

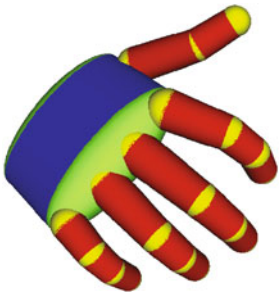


Fig. 1. Hand model with colored parts. Each color denotes a different type of geometric primitive (blue for elliptic cylinders, green for ellipsoids, yellow for spheres and red for cones).

2.1 Observation Model

The proposed hand pose recovery method operates on sequences of synchronized views acquired by intrinsically and extrinsically calibrated cameras. A set of images acquired from a set of such cameras at the same moment in time is called a *multiframe*. If $M_i = \{I_1, I_2, \ldots\}$ is a multiframe of a sequence $S = \{M_1, M_2, \ldots\}$ then I_j denotes the image from the j-th camera/view at the i-th time step. In the single camera case, a sequence of multiframes reduces to an image sequence.

An observation model similar to [10] is employed. For each image I of a multiframe M, an edge map $o_e(I)$ is computed by means of Canny edge detection [13] and a skin color map $o_s(I)$ is computed using the skin color detection method employed in [14]. As a convention, the label of 1 indicates presence and the label of 0 indicates the absence of skin or edges in the corresponding maps. For each edge map $o_e(I)$, a distance transform $o_d(I)$ is computed. For each image I, maps $O(I) = \{o_s(I), o_d(I)\}$ constitute its observation cues.

2.2 Hand Model

The model of hand kinematics used in this work is based on [15]. The kinematics of each finger, including the thumb, is modeled using four parameters encoding angles. More specifically, two are used for the base of the finger and two for the remaining joints. Bounds on the values of each parameter are set based on anatomical studies (see [15] and references therein). The global position of the hand is represented using a fixed point on the palm. The global orientation is parameterized using the redundant representation of quaternions. This parameterization results in a 26-DOF model encoded in a vector of 27 parameters.

The hand consists of a palm and five fingers. The palm is modeled as an ellipsoid cylinder and two ellipsoids for caps. Each finger consists of three cones and four spheres, except for the thumb which consists of two cones and three spheres (see Fig. 1). All required 3D shapes used in the adopted hand model consist of multiple instances of two basic geometric primitives, a sphere and a truncated cylinder. These geometric primitives, subjected to appropriate

homogeneous transformations, yield a model similar to that of [9]. Each transformation performs two different tasks. First, it appropriately transforms primitives to more general quadrics and, second, it applies the required kinematics. Using the shape transformation matrix

$$T_s = \begin{pmatrix} e \cdot s_x & 0 & 0 & 0 \\ 0 & e \cdot s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 1-e & e \end{pmatrix}, \tag{1}$$

spheres can be transformed to ellipsoids and cylinders to elliptic cylinders or cones. In Eq.(1), s_x, s_y and s_z are scaling factors along the respective axes. The parameter e is used only in the case of cones, representing the ratio of the small to the large radius of the cone before scaling (if not transforming to a cone, e is fixed to 1). Having a rigid transformation matrix T_k computed from the kinematics model, the final homogeneous transformation T for each primitive (sphere or cylinder) is

$$T = T_k \cdot T_s. \tag{2}$$

A non-trivial implementation issue (see Sec. 2.5) is the correct computation of surface normals. For given normals $\overrightarrow{n_i}$ of the two primitives in use, and given homogeneous transformation T, the computation of the new surface normals $\overrightarrow{n_i}'$ can be performed according to [16] using the equation $\overrightarrow{n_i}' = (T^{-\mathrm{T}})_{3\times3} \cdot \overrightarrow{n_i}$. $A_{3\times3}$ denotes the upper-left 3 by 3 submatrix of A.

Having a parametric 3D model of a hand, the goal is to estimate the model parameters that are most compatible to the observed images/image features (Sec. 2.1). To do so, we compute comparable image features from each hypothesized 3D hand pose (see Sec. 2.5). More specifically, given a hand pose hypothesis h, an edge map $r_e(h)$ and a skin color map $r_s(h)$ can be generated my means of rendering. The reference implementation of the rendering process is very similar to that of [9]. The informative comparison between each observation and corresponding hypotheses is detailed in Sec. 2.3.

2.3 Hypothesis Evaluation

The proposed method is based on a measure quantifying how compatible a given 3D hand pose is to the actual camera-based observations. More specifically, a distance measure between a hand pose hypothesis h and the observations of multiframe M needs to be established. This is performed by the computation of a function $E(h, M)$ which measures the discrepancies between skin and edge maps computed in a multiframe and the skin and edge maps that are rendered for a given hand pose hypothesis:

$$E(h, M) = \sum_{I \in M} D(I, h, \mathrm{C}(I)) + \lambda_k \cdot kc(h). \tag{3}$$

In Eq.(3), h is the hand pose hypothesis, M is the corresponding observation multiframe, I is an image in M, $\mathrm{C}(I)$ is the set of camera calibration parameters

corresponding to image I and λ_k is a normalization factor. The function D of Eq.(3) is defined as

$$D(I,h,c) = \frac{\sum o_s(I) \otimes r_s(h,c)}{\sum o_s(I) + \sum r_s(h,c) + \epsilon} + \lambda \frac{\sum o_d(I) \cdot r_s(h,c)}{\sum r_e(h,c) + \epsilon}, \tag{4}$$

where $o_s(I), o_d(I), r_s(h,c), r_e(h,c)$ are defined in Sec. 2.1. A small term ϵ is added to the denominators of Eq.4) to avoid divisions by zero. The symbol $\otimes$ denotes the logical XOR (exclusive disjunction) operator. Finally, λ is a constant normalization factor. The sums are computed over entire feature maps. The function kc adds a penalty to kinematically implausible hand configurations. Currently, only adjacent finger inter-penetration is penalized. Therefore, kc is defined as

$$kc(h) = \sum_{p \in \text{pairs}} \begin{cases} -\phi(p) & \phi(p) < 0 \\ 0 & \phi(p) \geq 0 \end{cases}, \tag{5}$$

where *pairs* denotes the three pairs of adjacent fingers, excluding the thumb, and ϕ denotes the difference between the abduction-adduction angles of those fingers. In all experiments the values of λ and λ_k were both set to 10.

2.4 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is an optimization technique that was introduced by Kennedy et al [17]. It is an evolutionary algorithm since it incorporates concepts such as populations, generations and rules of evolution for the atoms of the population (particles). A population is essentially a set of particles which lie in the parameter space of the objective function to be optimized. The particles evolve in runs which are called generations according to a policy which emulates "social interaction".

Canonical PSO, the simplest of PSO variants, was preferred among other optimization techniques due to its simplicity and efficiency. More specifically, it only depends on very few parameters, does not require extra information on the objective function (e.g., its derivatives) and requires a relatively low number of evaluations of the objective function [18]. Following the notation introduced in [19], every particle holds its current position (current candidate solution, set of parameters) in a vector x_t and its current velocity in a vector v_t. Moreover, each particle i stores in vector p_i the position at which it achieved, up to the current generation t, the best value of the objective function. Finally, the swarm as a whole, stores in vector p_g the best position encountered across all particles of the swarm. p_g is broadcasted to the entire swarm, so that every particle is aware of the global optimum. The update equations that are applied in every generation t to reestimate each particle's velocity and position are

$$v_t = K(v_{t-1} + c_1 r_1 (p_i - x_{t-1}) + c_2 r_2 (p_g - x_{t-1})) \tag{6}$$

and

$$x_t = x_{t-1} + v_t, \tag{7}$$

where K is a constant *constriction factor* [20]. In Eqs. (6), c_1 is called the *cognitive component*, c_2 is termed the *social component* and r_1, r_2 are random samples of a uniform distribution in the range [0..1]. Finally, $c_1 + c_2 > 4$ must hold [20]. In all performed experiments the values $c_1 = 2.8$, $c_2 = 1.3$ and $K = \frac{2}{\left|2-\psi-\sqrt{\psi^2-4\psi}\right|}$ with $\psi = c_1 + c_2$ were used.

Typically, the particles are initialized at random positions and their velocities to zero. Each dimension of the multidimensional parameter space is bounded in some range. If, during the position update, a velocity component forces the particle to move to a point outside the bounded search space, this component is zeroed and the particle does not perform any move at the corresponding dimension. This is the only constraint employed on velocities.

In this work, the search space is the 27-dimensional 3D hand pose parameter space, the objective function to be minimized is $E(M, h)$ (see Eq.(3)) and the population is a set of candidate 3D hand poses hypothesized for a single multiframe. Thus the process of tracking a hand pose requires the solution of a sequence of optimization problems, one for each of the acquired multiframes. By exploiting temporal continuity, the solution over multiframe M_t is used to generate the initial population for the optimization problem of M_{t+1}. More specifically, the first member of the population h_{ref} for M_{t+1} is the solution for M_t; The rest of the population consists of perturbations of h_{ref}. Since the variance of these perturbations depends on the image acquisition frame rate and the anticipated jerkiness of the observed hand motion, it has been experimentally determined in the reported experiments. The optimization for multiframe M_{t+1} is executed for a fixed amount of generations/iterations. After all generations have evolved, the best hypotheses h_{best} is dubbed as the solution for time step $t + 1$.

2.5 Exploiting Parallelism

A reference implementation of the proposed method was developed in MATLAB. A study of the computational requirements of the method components revealed that PSO and skin color detection are very fast. The computations of edge maps and their distance transforms are relatively slow but these tasks along with skin color detection are only executed once per multiframe. The identified computational bottlenecks are the rendering of a given 3D hand pose hypothesis and the subsequent evaluation of Eq.(3). More specifically, the hand model consists of a series of quadrics for which ray casting is used for rendering [9]. Additionally, since multiple quadrics overlap on the projection plane, pixel overwriting will occur and z-buffering is required so as to produce correct edge maps. The computation of Eq.(3) is a matter of pixel-wise multiplication and summation over entire images. The whole process is computationally expensive and prevents real-time performance. Reasonable PSO parameterizations where particles and generations range in the orders of tens, correspond to more than 4 minutes of processing time per multiframe.

GPU accelerated observation models have been employed in the past (e.g. [21]). In contrast to previous work, we provide a detailed description of a GPU implementation that exploits parallelism beyond the point of straightforward

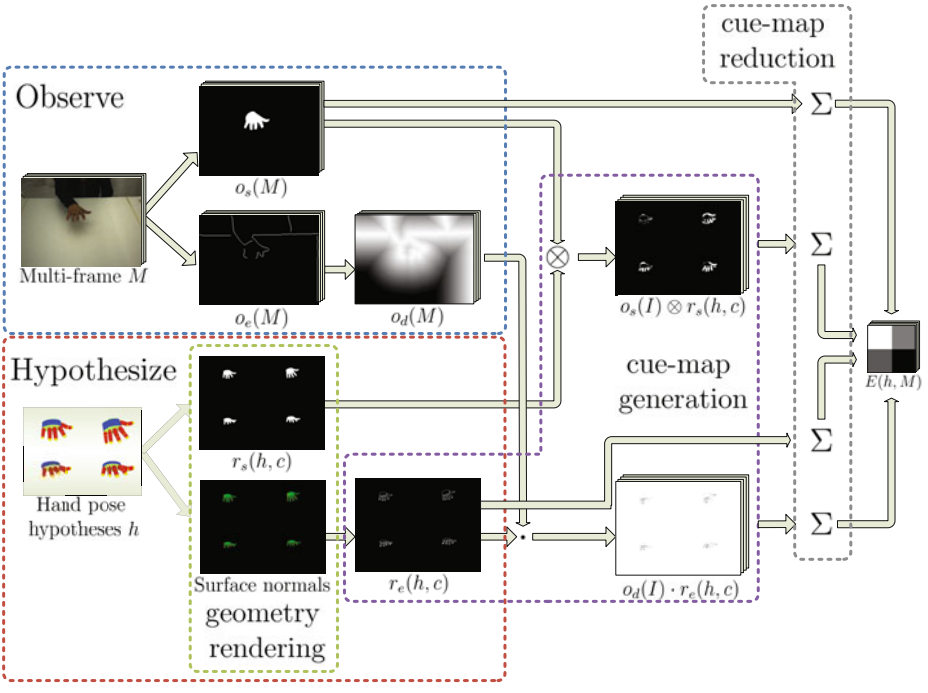


Fig. 2. Back-projection error computation flowchart. Observations of a human hand and hypothesized 3D poses are compared. Reference features are extracted from multiframes by means of skin color detection and edge detection. Artificial features are generated for the 3D pose hypotheses by means of rendering and edge detection. The three main GPU steps are annotated: geometry rendering, cue-map generation and cue-map reduction.

image processing and rendering. Our GPU implementation targets the acceleration of the two performance bottlenecks, i.e., rendering and evaluation. The rest of the tasks are also susceptible to acceleration (e.g. [22,23,24]) but this was not considered in this work. The final implementation used the Direct3D rendering pipeline to accelerate the computationally demanding tasks and MATLAB to perform the rest of the tasks as well as overall task coordination.

Rendering and evaluation of Eq.(3) are decomposed in three major GPU computation steps: geometry rendering, cue-map generation and cue-map reduction (see Fig. 2). Multiple particles are evaluated in large batches instead of single particles. This design choice defines a fine parallelization granularity which makes GPUs the optimal accelerator candidate.

Geometry rendering. The goal of the geometry rendering step is to simultaneously render multiple hand hypotheses in a big tiled rendering. Multiple renderings, instead of sequences of single renderings, were preferred in order to maximally occupy the GPU cores with computational tasks. The non-trivial issues to address are geometry instancing and multi-viewport clipping.

Hardware instancing [24] is used to perform multiple render batches efficiently. Efficiency regards both optimal GPU power exploitation and minimal memory usage. Batch rendering of multiple hand configurations essentially amounts to rendering of multiple instances of spheres and cylinders. However, the respective geometric instantiations are not required to be explicit. Hardware geometry instancing can be used in order to virtually replicate reusable geometry and thus make instantiation implicit.

A specialized pixel shader is used in order to perform custom multi-viewport clipping. Multiple viewports are required to be simultaneously rendered. However, conventional rendering pipelines do not account for multiple viewports, except for the case of sequential renderings. Unless multi-viewport clipping was performed, out of bounds geometry would expand beyond the tiles and spoil adjacent renderings.

The information that is transferred from CPU to GPU are the projection matrices c for each tile and the view matrix T for each primitive. The output of this rendering is the map $r_s(h, c)$, per pixel depth and per pixel normal vectors, encoded in four floating point numbers.

Cue-map generation. During cue-map generation, the output of the geometry rendering step is post-processed in order to provide cue-maps $r_s(h, c)$, $r_e(h, c)$, $o_s(I) \otimes r_s(h, c)$ and $o_e(I) \cdot r_e(h, c)$ of Eq.(3). Cue-map $r_s(h, c)$ passes through this stage since it is computed during geometry rendering (see Fig. 2). Cue-map $r_e(h, c)$ is computed by thresholding the discontinuity in normal vectors for a cross-neighborhood around each pixel. Cue-maps $o_s(I) \otimes r_s(h, c)$ and $o_e(I) \cdot r_e(h, c)$ are trivially computed by element wise operations between the operands.

Cue-map reduction. In the cue-map reduction step, scale space pyramids are employed to efficiently accumulate values across tiles. The expected input is an image that encodes maps $r_s(h, c)$, $r_e(h, c)$, $o_s(I) \otimes r_s(h, c)$ and $o_e(I) \cdot r_e(h, c)$ and the expected output is the sum over logical tiles of these maps. The pyramids are computed by means of sub-sampling, which is a very efficient GPU computation. Once the sums have been accumulated, the computation of Eq.(3) is straightforward.

3 Experimental Evaluation

The quantitative and qualitative experimental validation of the proposed method was performed based on both synthetic and real-world sequences of multiframes.

3.1 Quantitative Evaluation Based on Synthetic Data

The quantitative evaluation of the proposed method was based on synthetic sequences of multiframes which make possible the assessment of the proposed method against known ground truth. Towards this end, the hand model presented in Sec. 2.2 was animated so as to perform motions as simple as waving

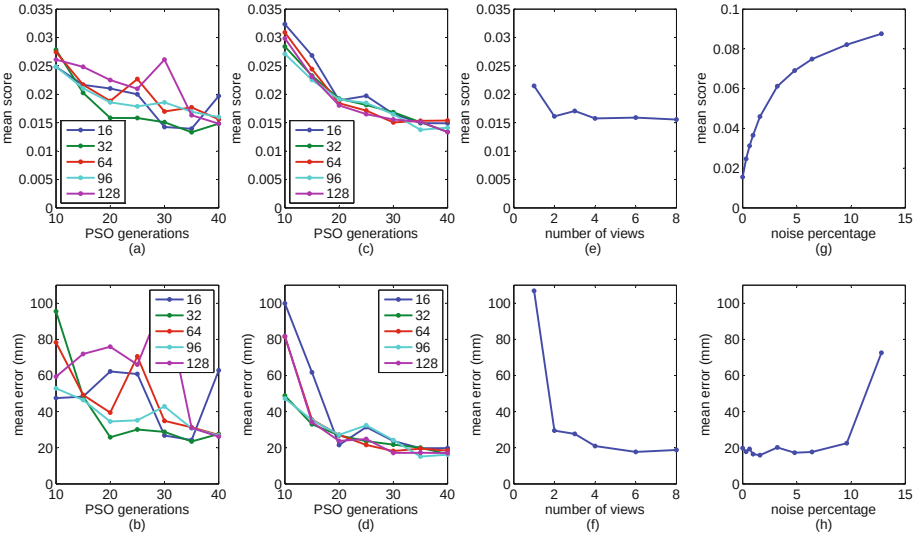


Fig. 3. Performance of the proposed method for different values of selected parameters. In the plots of the top row, the vertical axis represents the mean score E. In the plots of the bottom row, the vertical axis represents mean error in mm (see text for additional details). (a),(b): Varying values of PSO particles and generations for 2 views. (c),(d): Same as (a),(b) but for 8 views. (e),(f): Increasing number of views. (g),(h): Increasing amounts of noise.

and as complex as object grasping. A synthetic sequence of 360 poses of the moving hand was created. Each pose was observed by 8 virtual cameras surrounding the hand. This results in a sequence of 360 multiframes of 8 views, which constitute the input to the proposed method. The required cue maps were synthesized through rendering (see Sec. 2.2).

The performed quantitative evaluation assessed the influence of several factors such as PSO parameters, number of available views (i.e., multiframe size) and segmentation noise, over the performance of the proposed method. Figure 3 illustrates the obtained results. For each multiframe of the sequence, the best scoring hand pose h_{best} using the specified parameter values was found. Figures 3(a), (c), (e) and (g) provide plots of the score $E(h_{best}, M)$ (averaged for all multiframes M) as a function of various experimental conditions. Similarly, Figs. 3(b), (d), (f) and (h) illustrate the actual error in 3D hand pose recovery in millimeters, in the experimental conditions of Figs. 3(a), (c), (e) and (g), respectively. This error was computed as follows. The five fingertips as well as the center of the palm were selected as reference points. For each such reference point, the Euclidean distance between its estimated position and its ground truth position was first calculated. These distances were averaged across all multiframes, resulting in a single error value for the whole sequence.

Figures 3(a) and (b) show the behavior of the proposed method as a function of the number of PSO generations and particles per generation. In this

experiment, each multiframe consisted of 2 views with no noise contamination. It can be verified that varying the number of particles per generation does not affect considerably the error in 3D hand pose recovery. Thus, the number of generations appears to be more important than the number of particles per generation. Additionally, it can be verified that the accuracy gain for PSO parameterizations with more than 16 particles and more than 25 generations was insignificant. Figures 3(c), (d) are analogous to those of Figs 3(a),(b), except the fact that each multiframe consisted of 8 (rather than 2) views. The error variance is even smaller in this case as a consequence of the increased number of views which provides richer observations and, thus, more constraints. The accuracy gain for PSO parameterizations with more than 16 particles and more than 25 generations is even less significant.

In order to assess the behavior of the method with respect to the number of available views of the scene, experiments with varying number of views were conducted. Figures 3(e) and (f) show the behavior of the proposed method as a function of the size of a multiframe. For the experiments with less than 8 views, these were selected empirically so as to be as complementary as possible. More specifically, views with large baselines and viewing directions close to vertical were preferred. In these experiments, 128 PSO particles and 35 generations were used, and no segmentation noise was introduced in the rendered skin and edge maps. The obtained results (Figs. 3(e) and (f)) show that the performance improvement from one view to two views is significant. Adding more views improves the results noticeably but not significantly.

In order to assess the tolerance of the method to different levels of segmentation errors, all the rendered silhouette and edge maps were artificially corrupted with different levels of noise. The type of noise employed is similar to [7]. More specifically, positions are randomly selected within a map and the labels of all pixels in a circular neighborhood of a random radius are flipped. The aggregate measure of noise contamination is the percentage of pixels with swapped labels. In the plots of Figs. 3(g) and (h), the horizontal axis represents the percentage of noise-contaminated pixels in each skin map. Edge maps were contaminated with one third of this percentage. The contamination was applied independently to each artificial map r_s and r_e. In this experiment, 128 PSO particles and 35 PSO generations were used, and multiframes of eight views were considered.The plots indicate that the method exhibited robustness to moderate amounts of noise and failed for large amounts of noise. The exhibited robustness can be attributed to the large number of employed views. Since the noise of each view was assumed to be independent from all other views, the emerged consensus (over skin detection and edge detection) managed to cancel out low-variance noise. Figure 3 also demonstrates that the design choices regarding the objective function E (Sec. 2.3) are correct. This can be verified by the observed monotonic relation between E and the actual 3D hand pose estimation error.

Finally, Table 1 provides information on the runtime of these experiments. The table shows the number of multiframes per second for various parameterizations of the PSO (number of generations and number of particles per generation) and

Table 1. Number of multiframes per second processed for a number of PSO generations and camera views for 16/128 particles per generation

Generations	2 views	4 views	8 views
10	7.69/2.48	4.22/1.26	2.14/0.63
15	7.09/1.91	3.65/0.97	1.85/0.49
20	**6.23**/1.55	3.19/0.79	1.62/0.39
25	5.53/1.31	2.85/0.67	1.44/0.33
30	5.00/1.13	2.59/0.57	1.30/0.29
35	4.55/1.00	2.34/0.50	1.18/0.25
40	4.18/0.89	2.15/0.45	1.09/0.23

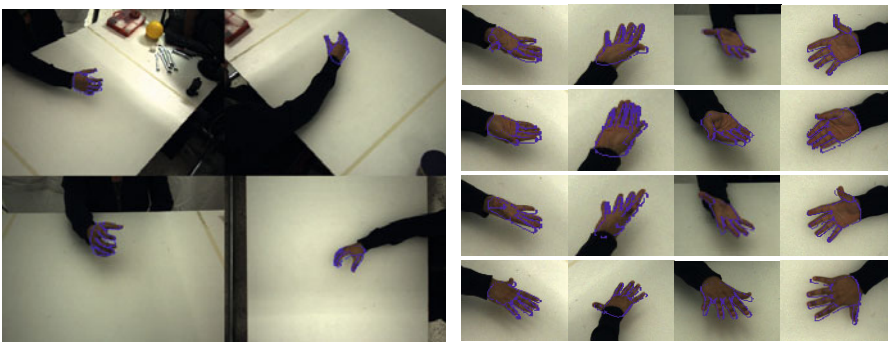


Fig. 4. Sample frames from real-world experiments. Left: four views of a multiframe of a cylindrical grasp. Right: Zoom on hands; Rows are from the same multiframe and columns correspond to the same camera view.

various number of views. The entry in boldface corresponds to 20 generations, 16 particles per generation and 2 views. According to the quantitative results presented earlier, this setup corresponds to the best trade-off between accuracy of results, computational performance and system complexity. This figure shows that the proposed method is capable of accurately and efficiently recovering the 3D pose of a hand observed from a stereo camera configuration at 6.2Hz. If 8 cameras are employed, the method delivers poses at a rate of 1.6Hz.

3.2 Experiments with Real World Images

Real-world image sequences were acquired using a multicamera system which is installed around a $2 \times 1m^2$ bench and consists of 8 *Flea2* PointGrey cameras. Cameras are synchronized by a timestamp-based software that utilizes a dedicated *FireWire 2* interface ($800\,MBits/sec$) which guarantees a maximum of $125\,\mu sec$ temporal discrepancy in images with the same timestamp. Each camera has a maximum framerate of 30 fps at highest (i.e. 1280×960) image resolution. The workstation where images are gathered has a quad-core Intel i7 920 CPU, 6 GBs RAM and an Nvidia GTX 295 dual GPU with $894\,GFlops$ processing power and 896 MBs memory per GPU core.

Several sequences of multiframes have been acquired, showing various types of hand activities such as isolated motions and hand-environment interactions including object grasping. Figure 4 provides indicative snapshots of 3D hand pose estimation superimposed on the original image data. Videos with results of these experiments are available online[1].

4 Discussion

In this paper, we proposed a novel method for the visual recovery of 3D hand pose of a human hand. This is formulated as an optimization problem which is accurately and robustly solved through Particle Swarm Optimization. In an effort to propose a method that is both accurate and computationally efficient, appropriate design choices were made to select components that exhibit data parallelism which is exploited by a GPU based implementation. The experimental evaluation in challenging datasets (complex hand articulation, distant hand views) demonstrates that accurate pose recovery can be achieved at a framerate that greatly outperforms the current state of the art. The individual constituents of the proposed method are clearly separated. It is quite easy for changes to be made to the objective function, the optimization method or the hand model without affecting the other parts. Current research if focused on considering more compact search spaces through the use of dimensionality reduction techniques.

Acknowledgements. This work was partially supported by the IST-FP7-IP-215821 project GRASP. The contributions of Asimina Kazakidi and Thomas Sarmis (members of the CVRL/ICS/FORTH) are gratefully acknowledged.

References

1. Moeslund, T.B., Hilton, A., Krüger, V.: A survey of advances in vision-based human motion capture and analysis. CVIU 104, 90–126 (2006)
2. Wang, R.Y., Popović, J.: Real-time hand-tracking with a color glove. ACM Transactions on Graphics 28, 1 (2009)
3. Erol, A., Bebis, G., Nicolescu, M., Boyle, R.D., Twombly, X.: Vision-based hand pose estimation: A review. CVIU 108, 52–73 (2007)
4. Athitsos, V., Sclaroff, S.: Estimating 3d hand pose from a cluttered image. In: CVPR, vol. 2, p. 432 (2003)
5. Rosales, R., Athitsos, V., Sigal, L., Sclaroff, S.: 3d hand pose reconstruction using specialized mappings. In: ICCV, pp. 378–385 (2001)
6. Wu, Y., Huang, T.S.: View-independent recognition of hand postures. In: CVPR, pp. 88–94 (2000)
7. Romero, J., Kjellstrom, H., Kragic, D.: Monocular real-time 3D articulated hand pose estimation. In: IEEE-RAS Int'l Conf. on Humanoid Robots, pp. 87–92 (2009)
8. Rehg, J.M., Kanade, T.: Visual tracking of high dof articulated structures: An application to human hand tracking. In: Eklundh, J.-O. (ed.) ECCV 1994. LNCS, vol. 801, pp. 35–46. Springer, Heidelberg (1994)

[1] http://www.ics.forth.gr/∼argyros/research/3Dhandpose.htm

9. Stenger, B., Mendonca, P., Cipolla, R.: Model-based 3D tracking of an articulated hand. In: CVPR, pp. II–310–II–315 (2001)
10. Sudderth, E., Mandel, M., Freeman, W., Willsky, A.: Visual hand tracking using nonparametric belief propagation. In: CVPR Workshop, pp. 189–189 (2004)
11. de la Gorce, M., Paragios, N., Fleet, D.: Model-based hand tracking with texture, shading and self-occlusions. In: CVPR, pp. 1–8 (2008)
12. John, V., Trucco, E., Ivekovic, S.: Markerless human articulated tracking using hierarchical particle swarm optimisation. Image and Vision Computing 28, 1530–1547 (2010)
13. Canny, J.: A computational approach to edge detection. PAMI 8, 679–698 (1986)
14. Argyros, A., Lourakis, M.: Real-time tracking of multiple skin-colored objects with a possibly moving camera. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3023, pp. 368–379. Springer, Heidelberg (2004)
15. Albrecht, I., Haber, J., Seidel, H.: Construction and animation of anatomically based human hand models. In: 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Eurographics Association, p. 109 (2003)
16. Turkowski, K.: Transformations of surface normal vectors. Technical report, Tech. Rep. 22, Apple Computer (July 1990)
17. Kennedy, J., Eberhart, R., Shi, Y.: Swarm intelligence. Morgan Kaufmann Publishers, San Francisco (2001)
18. Angeline, P.: Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences. In: Porto, V.W., Waagen, D. (eds.) EP 1998. LNCS, vol. 1447, pp. 601–610. Springer, Heidelberg (1998)
19. White, B., Shaw, M.: Automatically tuning background subtraction parameters using particle swarm optimization. In: IEEE ICME, pp. 1826–1829 (2007)
20. Clerc, M., Kennedy, J.: The particle swarm - explosion, stability, and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation 6, 58–73 (2002)
21. Shaheen, M., Gall, J., Strzodka, R., Gool, L.V., Seidel, H.P.: A comparison of 3d model-based tracking approaches for human motion capture in uncontrolled environments. In: Workshop on Applications of Computer Vision, pp. 1–8 (2009)
22. Luo, Y., Duraiswami, R.: Canny edge detection on NVIDIA CUDA. In: CVPR 2008 Workshops, pp. 1–8 (2008)
23. Fischer, I., Gotsman, C.: Fast approximation of high-order Voronoi diagrams and distance transforms on the GPU. Journal of Graphics, GPU, & Game Tools 11, 39–60 (2006)
24. Pharr, M., Fernando, R.: Gpu gems 2: programming techniques for high-performance graphics and general-purpose computation (2005)

Full pose estimation of a hand interacting with objects: Exploiting context to turn occlusions into a useful visual cue

Iasonas Oikonomidis[b,a], Nikolaos Kyriazis[b,a], Antonis Argyros[1 b,a]

[a] *Institute of Computer Science, FORTH, Heraklion, Crete, Greece*
[b] *Computer Science Department, University of Crete, Greece*

Abstract

Due to occlusions, the estimation of the full pose of a human hand interacting with an object is much more challenging than pose recovery of a hand observed in isolation. In this work we formulate an optimization problem whose solution is the 26-DOF hand pose together with the model parameters and pose of the manipulated object, that jointly best explain the incompleteness of hand observations resulting from occlusions due to hand-object interaction. Thus, occlusions is not a curse we bypass but a feature we exploit. The proposed method is the first that provides accurate and fast solution to this problem. Additionally, it is the first to demonstrate that hand-object interaction is not necessarily a complicating factor but a context that can be exploited effectively for hand pose estimation. Extensive quantitative and qualitative experiments with simulated and real world image sequences as well as a comparative evaluation with a state-of-the-art method for pose estimation of isolated hands, support the above findings.

Key words: Multiple objects tracking, Occlusion, Object permanence

1. Introduction

The estimation of the full pose of hands from markerless visual observations is a problem whose solution is of fundamental importance in numerous applications including but not limited to the visual perception of grasping and manipulation, sign language understanding, etc. As it is common to many interesting problems, a lot of challenges are associated with it. A number

of cascading issues such as the dimensionality of the problem, the incomplete and/or ambiguous observations due to scene clutter and the requirement for accurate estimates in real time, hinder a practical and effective solution.

Full DOF hand pose recovery during hand-object interaction is an even more difficult problem due to the induced hand-object occlusions. We aim at exploiting contextual information to benefit from these occlusions. In a hand-object interaction scenario, the hand *together with* the object constitute indispensable components of an integral context. Intuition suggests that any effort to estimate the state of the hand or of the object in isolation, is bound to be suboptimal, exactly because it does not exploit the whole spectrum of available information.

In this work, it is assumed that hand-object interaction is observed by a multicamera system. In each of the acquired views, edge and skin color maps form 2D cues of the presence of a hand. The presence of an object and the associated object-hand occlusions result in missing observations of the performing hand. As an example, consider the situation depicted in Fig. 1. Depending on the viewpoint and the actual hand-object configuration, certain parts of the hand are occluded. Clearly, this incomplete observation of the hand provides important evidence on the type and pose of the manipulated object. Conversely, attributing missing hand observations (skin color, hand edges) to the presence of a manipulated object permits a more accurate estimation of the pose of the partially observed hand. The tight coupling between "what the hand tells about the object" and "what the objects tells about the hand", suggests that we should identify *simultaneously* the hand configuration and the object 3D model and pose that best explain the observed scene holistically. We exploit this coupling by formulating an optimization problem whose solution is the full DOF hand pose and the object model and pose that best explain the available hand-object observations but also the lack of them. In that sense, occlusions are turned into a cue that contributes effectively to the solution of the problem.

1.1. Related work

The recovery of the full 3D structure of articulated objects such as humans and hands presents a lot of challenges. Several approaches have been proposed that address various aspects of the problem such as its dimensionality, the incomplete and/or ambiguous observations due to scene

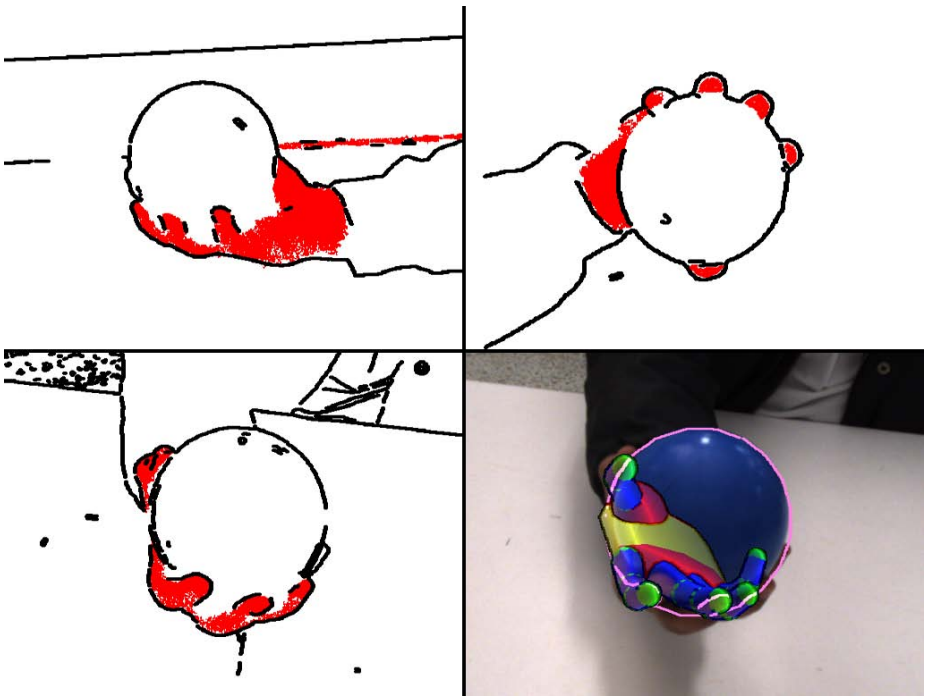


Figure 1: Top row, and bottom left: Three views of a hand grasping an object. Skin regions appear in red and edges in black. The hand is partially occluded by the object in all views. The incomplete skin and edge maps of the hand facilitate the generation of a hypothesis for a hand manipulating a compact sphere. At the same time, given this hypothesis, the 3D pose of the hand can be estimated more accurately. Bottom right: the output of the proposed approach superimposed in one of the frames.

clutter, its computational requirements, etc. Moeslund et al. [12] provide a review of research to the general problem of visual human motion capture and analysis. A review that is specific to the problem of human hand motion estimation is provided in [7]. Related methods can be categorized as partial or full pose estimation methods, depending on the level of detail they provide regarding the observed hand.

Another categorization identifies appearance-based and model-based methods. Appearance-based methods estimate hand configurations by learning a direct mapping of image features to the hand configuration space [3, 19, 23, 18]. Model-based approaches employ a 2D or a 3D hand model [16, 20, 21, 6, 13]. In the case of 3D hand models the hand pose is estimated by matching the projection of the model to the observed image features. The task is then formulated as a search problem in a high dimensional configuration space, which typically induces a high

computational cost. A common characteristic of all the methods mentioned above is that they consider human hands in isolation. Thus, in the context of hand-object interaction, their accuracy in hand pose estimation is compromised due to the induced hand-object occlusions that affect drastically the completeness of hand observations.

Given the significant role of context in human visual recognition [14], several researchers have attempted to exploit contextual constraints in solving computer vision problems. For example Rabinovich et al. [15] exploit scene context in the problem of object detection while Marszalek et al. [11] do the same for understanding actions. More closely related to our problem, a few recent works [10, 8, 24, 25, 17] consider context for classifying human-object interaction activities. The related methods can be classified based on whether they refer to the human body or hand and also on whether they provide a detailed 3D model of the actor (human body or hand) and the object. Thus, [8, 24, 25] study the full human body while in interaction with objects. From these, only [25] provides detailed information on human body pose. Kjellstrom et al. [10] consider hand-object interactions but only for classifying them, without providing a detailed 3D human hand and object model. Finally, Romero et al. [17] propose an appearance based method for estimating the pose of a hand interacting with an object. However, occlusions are treated as a complicating factor that needs to be tolerated and no information on the object is provided. A method that exploits context to provide a detailed 3D model for both hands and objects seems to be missing from the current literature. The proposed method is trying to fill this gap.

Towards this direction, in this work we extend the approach in [13] by considering jointly the hand and the manipulated object. In [13], a generative, multiview method for 3D hand pose recovery is presented. In each of the acquired views, reference features are computed based on skin color and edge detection. A 26-DOF 3D hand model is adopted. For a given hand configuration, skin and edge feature maps are rendered and compared directly to the respective observations. The discrepancy of a given 3D hand pose to the observations is quantified by an objective function that is minimized through Particle Swarm Optimization (PSO). The whole approach is implemented efficiently on a GPU. In the proposed approach, we do not only seek for the optimal hand model that explains the available hand observations but rather the joint hand-

object model that best explains both the available and the missing hand/object observations. It is demonstrated that the aforementioned conceptual difference is very important in solving more accurately a more complex and interesting problem.

To the best of our knowledge, the proposed method is the first to demonstrate that hand-object interaction is not necessarily a complicating factor towards estimating the configuration of a hand. On the contrary, this contextual information can be exploited effectively towards solving the problem more accurately. Additionally, it is the first model based method that provides full, accurate 26-DOF hand pose estimation during hand-object interaction. As a valuable additional result, the method provides a parametric 3D model of the manipulated object together with its 3D position and orientation. This is achieved by exploiting the hand-object occlusions and despite the lack of an explicit object appearance model. The approach explores an essentially infinite configuration space. Thus, its accuracy is not limited by the size and content of the database of hand configurations, as e.g. in [18]. The above findings are supported by qualitative and quantitative experiments with both simulated and real world image sequences as well as by a comparative evaluation with the method in [13].

2. Hand-object pose estimation (*HOPE*)

The problem of joint hand-object pose estimation is formulated as a multidimensional optimization problem. In the following, we present in detail the basic building blocks of the proposed method for joint Hand-Object Pose Estimation (*HOPE*), with emphasis on the employed observation model, joint hand-object 3D model, hypothesis evaluation mechanism and optimization method.

2.1. Computed visual cues

The proposed method operates on sequences of synchronized views acquired by intrinsically and extrinsically calibrated cameras. A set of images acquired from these cameras at the same moment in time is called a *multiframe*. If $M_i = \{I_1, I_2, \ldots\}$ is a multiframe of a sequence $S = \{M_1, M_2, \ldots\}$, I_j denotes the image from the j-th camera/view at the i-th time step. For each

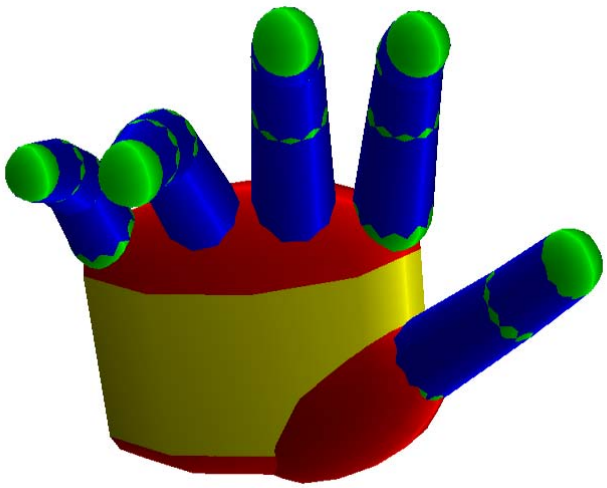


Figure 2: Graphical illustration of the employed 26-DOF 3D hand model, consisting of 37 geometric primitives. The same type of geometric primitives appear in the same color (yellow for elliptic cylinders, red for ellipsoids, green for spheres and blue for cones).

image I of a multiframe M, an edge map $o_e(I)$ is computed through Canny edge detection [4] and a skin color map $o_s(I)$ is computed using the method presented in [2]. As a convention, the label 1 indicates presence and the label 0 indicates absence of skin or edges in the respective maps. For each edge map $o_e(I)$, a distance transform $o_d(I)$ is computed. For each image I, maps $O(I) = \{o_s(I), o_d(I)\}$ constitute its observation cues.

2.2. Joint hand-object model

The proposed approach employs a model $m = (h, o)$ that represents jointly a hand h and the manipulated object o. The hand model h consists of a palm and five fingers. The palm is modeled as an ellipsoid cylinder and two ellipsoids for caps. Each finger consists of three cones and four spheres, except for the thumb that consists of two cones, an ellipsoid and three spheres (see Fig. 2). The resulting total of 37 3D geometric primitives of the hand model are different parameterizations of an ellipsoid and a truncated cylinder. The assembly of appropriate homogeneous transformations of these two geometric primitives yield a hand model similar to that of [20].

Hand kinematics is based on [1]. The kinematics of all five fingers is modeled using four parameters encoding angles, two for the base of the finger and two for the remaining joints. Ranges of parameter values are determined based on anatomical studies [1]. The position of a fixed point on the palm defines the global position of the hand. The global orientation is parameterized using the redundant representation of quaternions. This parameterization results in a 26-DOF model encoded in a vector of 27 parameters.

For representing an object, in principle, any parametric model o can be used. The representation of the most common handheld objects such as cuboids, ellipsoids and cylinders requires 3, 3 and 2 intrinsic shape parameters, respectively. More complex parametric shape models like superquadrics require as many as 6 parameters. Regardless of the intrinsic shape parameterization, seven additional parameters are required, 3 for 3D position and 4 for a quaternion-based representation of 3D orientation. In this work, we provide experimental results with ellipsoids, cuboids and cylinders. Nevertheless, there is no inherent limitation that prevents the method from being able to handle more complex object models, provided that this does not increase the dimensionality of the problem prohibitively.

The joint hand-object model $m = (h, o)$ consists of the concatenation of the parameters of the hand and the object model. Interestingly, some of the model parameters are coupled. As a concrete example, the global position of the hand and the global position of the object are coupled in a hand-object interaction scenario, in the sense that the object cannot be located arbitrarily far from the object.

2.3. Evaluation of hand-object model hypotheses

Given a joint parametric hand-object model $m = (h, o)$, the goal is to estimate the parameters that are most compatible to the observed image features (Sec. 2.1). To do so, we first compute comparable image features from each hypothesized hand-object model. More specifically, an edge map $r_e(m)$ and a skin color map $r_s(m)$ can be generated my means of rendering. The reference implementation of the rendering process is very similar to that of [20]. The implicit assumption made at this point is that, although the appearance of the object is unknown, it cannot contain skin-colored pixels. Thus, the hand component h of m contributes to the skin color map

$r_s(m)$ by setting visible hand pixels to 1, while the object component o of m contributes to the skin color map $r_s(m)$ by setting map pixels to 0. Experimental results have verified that the presence of a moderate number of skin-colored pixels on the object's surface does not affect the accuracy of the method.

Hypothesis evaluation is based on a measure quantifying the compatibility of a given hand-object model m to the actual camera-based observations. Towards this end, a distance measure between a hand-object hypothesis m and the observations of multiframe M needs to be established. Similarly to [13], this is performed by the computation of a function $O(m, M)$ which measures the discrepancies between the skin and edge maps computed in a multiframe M and the skin and edge maps that are rendered for m:

$$O(m, M) = \sum_{I \in M} D(I, m, \mathrm{C}(I)) + \lambda_k W(h). \tag{1}$$

In Eq.(1), I is an image in M, $\mathrm{C}(I)$ is the set of camera calibration parameters corresponding to view/image I and λ_k is a normalization factor. The function D is defined as

$$\begin{aligned} D(I, m, c) \quad = \quad & \frac{\sum o_s(I) \otimes r_s(m, c)}{\sum o_s(I) + \sum r_s(m, c) + \epsilon} \\ & + \lambda \frac{\sum o_d(I) \cdot r_s(m, c)}{\sum r_e(m, c) + \epsilon}, \end{aligned} \tag{2}$$

where $o_s(I), o_d(I), r_s(m, c), r_e(m, c)$ are defined in Sec. 2.1. A small term ϵ is added to the denominators of Eq.(2) to avoid divisions by zero. The symbol $\otimes$ denotes the pixelwise logical XOR of its binary image mask operands. Finally, λ is a constant normalization factor. In all experiments the values of the normalization factors λ and λ_k were set to 10. The sums are computed over entire feature maps. The function W adds a penalty to kinematically implausible hand configurations such as adjacent finger inter-penetration. Thus, W is defined as

$$W(h) = \sum_{p \in \mathrm{P}} \begin{cases} -\phi(p) & \phi(p) < 0 \\ 0 & \phi(p) \geq 0 \end{cases}, \tag{3}$$

where P denotes the three pairs of adjacent fingers, excluding the thumb, and ϕ denotes the difference between the abduction-adduction angles of those fingers. Ideally, function W should also penalize hand-object configurations that are kinematically impossible in the sense that hand parts share the same physical space with the manipulated object. Although this is a possibly valuable constraint, it is not exploited in the current implementation of *HOPE*.

2.4. Optimization

The minimization of the objective function of Eq.(1) is achieved through Particle Swarm Optimization (PSO). Introduced by Kennedy et al. [9], PSO achieves optimization through a policy which emulates "social interaction" of a population of atoms (particles) that evolves in a number of generations. A population is essentially a set of particles that lie in the parameter space of the objective function to be optimized.

Following the notation introduced in [22], every particle holds its current position (current candidate solution, set of parameters) in a vector x_t and its current velocity in a vector v_t. Moreover, each particle i stores in vector p_i the position which corresponds to the best evaluation of its objective function up to the current generation t. Finally, the swarm as a whole, stores in vector p_g the best position encountered across all particles of the swarm. p_g is broadcasted to the entire swarm, so that every particle is aware of the global optimum. In every generation t, the velocity of each particle is updated according to

$$v_t = K(v_{t-1} + c_1 r_1 (p_i - x_{t-1}) + c_2 r_2 (p_g - x_{t-1})) \tag{4}$$

and its position x_t according to

$$x_t = x_{t-1} + v_t. \tag{5}$$

In the above equations, K is a constant constriction factor [5], c_1 is called the cognitive component, c_2 is termed the social component and r_1, r_2 are random samples of a uniform distribution in [0..1]. Finally, $c_1 + c_2 > 4$ must hold [5]. In all performed experiments the values $c_1 = 2.8$, $c_2 = 1.3$ and $K = 2/|2 - \psi - \sqrt{\psi^2 - 4\psi}|$ with $\psi = c_1 + c_2$ were used.

The search space is a multidimensional cuboid. The particle positions are initialized randomly and the particle velocities are set to zero. If, during the position update, a velocity component forces the particle to move outside the search space, this component is zeroed and the particle does not perform any move at the corresponding dimension. The final outcome of the PSO is the model parameters p_*, i.e., the particle with the best score across all generations.

The search space of *HOPE* is the joint hand-object model parameter space m. Given a hand model represented by 27 parameters and an object model represented by d parameters, the overall problem needs to be solved in a $(27 + d)$-dimensional space. The objective function to be minimized is $O(m, M)$ (see Eq.(1)) and the population is a set of candidate 3D hand-object configurations hypothesized for a single multiframe. The resulting solution $m_* = (h_*, o_*)$ represents the best guess of the algorithm for the joint hand and object model.

To cope with the tracking of the hand-object configuration in time, a series of optimization problems needs to be solved, one for each of the acquired multiframes. By exploiting temporal continuity, the solution for multiframe M_{t-1} is used to generate the initial population for the optimization problem of M_t. More specifically, the first member of the population m_{ref} for M_t is the solution for M_{t-1}; The rest of the population consists of perturbations of m_{ref}. The optimization for multiframe M_t is executed for a fixed amount of generations/iterations. After all generations have evolved, the best hypothesis m_* is dubbed as the solution for time step t.

3. Experimental Evaluation

The proposed method has been validated extensively based on both synthetic and real-world sequences of multiframes. First, we demonstrate the accuracy and the computational performance of the proposed hand-object pose estimation (*HOPE*) method on a synthetically rendered data set where hands perform different grasps on a variety of objects (see Sec. 3.1). We also compare the performance of *HOPE* to that of the method in [13], hereafter abbreviated as *PEHI* (Pose Estimation of Hands in Isolation). A final experiment with synthetic data involves the application of *HOPE* to a data set showing hands in isolation. The goal of this experiment is to show that *HOPE* can also estimate the pose of hands in isolation effectively, as a special case.

Besides the synthetic data, we also provide qualitative evidence on how the *HOPE* and *PEHI* algorithms perform on real sequences of multiframes (see Sec. 3.2). Although ground truth information is not available for these sequences, indicative results confirm the superiority of *HOPE* over *PEHI* which is in accordance with the experimental results over synthetic data.

3.1. Experiments on synthetic data

Experiments with synthetically produced sequences of multiframes were performed to make possible the assessment of the proposed method based on ground truth data. To that end, we simulated different grasps of three different objects (an ellipsoid, a cylinder, and a box) performed by the employed hand model (Sec. 2.2). The interaction of the hand with each of these three objects was observed by 8 virtual cameras surrounding the scene. This resulted in three sequences consisting of 116 multiframes of 8 frames, each. The required cue maps (edges, skin color) were synthesized through rendering (see Sec. 2.2).

For the quantitative evaluation of the method, an error metric quantifying the discrepancy between a true hand pose and an estimated hand pose is required. This metric was computed as follows. The five fingertips as well as the center of the palm were selected as reference points. For each such reference point, the Euclidean distance between its estimated position and its ground truth position is first calculated. For a given set of PSO parameters, these distances are averaged across all multiframes of each sequence, and all sequences. This results in a single error value $\mathcal{D}$ for the whole dataset under the specific PSO parameters.

Figures 3(a), (b) and (c) illustrate the estimated error $\mathcal{D}$ of the *HOPE* method as a function of the PSO parameters and the number of available views (i.e., multiframe size). In Fig. 3(a), $\mathcal{D}$ is plotted as a function of the number of PSO generations and particles per generation, for multiframes consisting of 2 views. $\mathcal{D}$ takes values between 13*mm* and 65*mm*. It can be verified that for more that 30 PSO generations and more than 32 particles per generation the error in 3D hand pose recovery for *HOPE* does not vary considerably and it is in the order of 15*mm*.

Figure 3(d) is analogous to that of Fig. 3(a) for the *PEHI* algorithm. In this case, the mean error $\mathcal{D}$ does not decrease monotonically as a function of particles. This is attributed to the incomplete/occluded hand observations that undermine the convergence of *PEHI*. $\mathcal{D}$ now ranges

Table 1: Estimated/actual parameters for the object models in the experiments with synthetic data.

Object	Estimated/Actual parameters (in *mm*)
Cylinder	Radius: 127/128, Height: 54/55
Ellipsoid	X: 55/55, Y: 86/85, Z: 129/128
Box	X: 77/77, Y 128/129, Z: 153/156

between 55*mm* and 120*mm*. It can be verified that for more that 30 PSO generations and more than 32 particles per generation the error in 3D hand pose recovery for *PEHI* is in the order of 80*mm*. For these parameters, the error of *PEHI* is on average 4.75 times larger than that of *HOPE*, thus *HOPE* clearly outperforms *PEHI*.

Figures 3(b) and (e) are similar to those of Figs. 3(a) and (d), except the fact that each multiframe now consists of 8 rather than 2 views. $\mathcal{D}$ takes values between 8*mm* and 60*mm* for *HOPE* and between 11*mm* and 76*mm* for *PEHI*. For more that 30 PSO generations and more than 32 particles per generation the error of *PEHI* is 1.4 times larger than that of *HOPE*, thus *HOPE* still performs considerably better. However, the addition of more views, some of which provide more complete observations of hand-object interaction, narrows down the difference in the performance of the two algorithms.

In order to assess the behavior of the method with respect to the number of available views of the scene, additional experiments with a varying number of views were conducted. Figures 3(c) and (f) show the behavior of *HOPE* and *PEHI* as a function of the size of a multiframe. For the experiments with less than 8 views, these were selected empirically to be as different as possible. PSO optimization involved 64 particles running for 40 generations. The obtained results show that the performance improvement from one view to two views is significant for both algorithms. In all cases, *HOPE* performs better than *PEHI*.

Overall, the experiments in Fig. 3 show a consistent superiority of *HOPE* over *PEHI* which is dominant in the case of a limited number of available views. This is important because it allows for a practical joint hand-pose estimation by a multicamera system consisting of much less cameras and thus in a system with less costs, complexity and requirements for computational resources.

Table 2: Average multiframe processing times (in sec) for *HOPE* and *PEHI* running for 40 generations, 64 particles/generation and varying number of multiframe sizes.

Algorithm	2 cameras	4 cameras	8 cameras
HOPE	1.06	2.10	4.35
PEHI	1.02	2.05	4.13

Besides its superiority in hand pose estimation, *HOPE* also estimates the model parameters of the manipulated object. The average positional error of object detection across all sequences of multiframes in the experiments of Fig. 3 is $2mm$ (Euclidean distance between true position and estimated one) and the average orientation error is 3 deg. Table 1 shows the actual and estimated object parameters. The later are averaged for all the multiframes of the sequence that depicts the corresponding object. It can be verified that for all types of objects, the estimated model parameters are very close to the ground truth.

Table 2 summarizes the per multiframe processing time[2] for different multiframe sizes and for runs of 40 PSO generations and 64 particles per generation for both *HOPE* and *PEHI*. Each entry is the average value over 116 multiframes. For both algorithms, the processing time is almost linear to the number of views. It can also be observed that *HOPE* is only $2\% - 5\%$ slower than *PEHI*.

Finally, we applied both *HOPE* and *PEHI* to a synthetic image sequence (400 multiframes, 8 frames/multiframe) showing non-rigid motion of hands in isolation. Figure 4 plots the mean error $\mathcal{D}$ as a function of the number of the employed views. For both algorithms, 40 PSO generations and 64 particles per generation were used. For *HOPE*, a cylindrical object has been hypothesized. The result shows that the performance of the two algorithms is comparable, a fact that indicates the capability of *HOPE* to track hands observed in isolation. Expectedly, *HOPE* estimated the presence of very small objects (size in the order of a few *mm*s).

3.2. Experiments on real image data

Real-world image sequences were acquired using a multicamera system (Fig. 5) installed around a $2\times1m^2$ bench and consisting of 8 synchronized and calibrated *Flea2* PointGrey cameras.

[2]Experiments run on the computational infrastructure presented in Sec.3.2.

Table 3: Estimated/actual parameters for the object models in the experiments of Fig. 6.

Object	Estimated/actual parameters (in *mm*)
Cylinder	Radius: 54/53, Height: 118/131
Ellipsoid	X: 127/116, Y: 127/116, Z: 123/116
Box	X: 71/67, Y 145/150, Z: 97/93

Each camera has a maximum framerate of 30 fps, at 1280×960 image resolution. However, the core processing is performed on 256×256 windows centered around the rendered hand-object edges of the previous multiframe solution. The workstation where images are gathered and processed is equipped with a quad-core Intel i7 920 CPU, 6 GBs RAM and an Nvidia GTX 295 dual GPU with 894 $GFlops$ processing power and 896 MBs memory per GPU core.

Three sequences of multiframes have been acquired, each showing a hand grasping and manipulating a spherical (301 multiframes), a cylindrical (261 multiframes), and a box (251 multiframes) object. Figure 6(a) provides sample results obtained by applying *HOPE* (top row) and *PEHI* (bottom row) to a specific multiframe of the sphere sequence. Since the hand is mostly occluded by the sphere in all views, *HOPE* estimates the hand configuration correctly while *PEHI* fails completely. Similar results were obtained in the case of the cylinder sequence which shows a hand grasping and turning up-side down a cylindrical object. Instead of providing sample views of a single multiframe, Fig. 6(b) shows four frames acquired from the same camera in different moments in time. *HOPE* manages to track the configuration of the hand throughout the whole sequence while *PEHI* looses track of the hand as soon as the later becomes severely occluded by the object. Figure 6(c) shows a similar result for the box sequence.

The lack of ground truth information does not permit a quantitative assessment of the accuracy in hand pose estimation on these sequences. However, in Table 3, we compare the object shape parameters estimated by *HOPE* to the actual, physically measured ones. The estimated parameters are averaged for all multiframes of a given sequence. The standard deviation of these parameters is in the order of a few millimeters. It can be verified that the error in object shape estimation is satisfactory.

For *HOPE*, we also run a simple classification experiment. More specifically, for the sphere

Table 4: The mean value of the objective function of *HOPE* and its standard deviation when optimization searches for cylinders, ellipsoids and cuboids for a sequence showing an ellipsoid (sphere).

	Cylinder	Ellipsoid	Cuboid
Mean value	0.41	0.35	0.48
Stdev.	0.09	0.08	0.14

sequence (see Fig. 6(a) and (b)), we run *HOPE* assuming a cuboid, an ellipsoid and a cylinder. Table 4 shows the mean value and the standard deviation of the objective function of *HOPE* in all the multiframes of the sequence. As it can be verified, the hypothesis of an ellipsoid better explains the observed scene. In fact, 90.5% of the multiframes were better explained by the ellipsoid, 9.5% by the cylinder and none by the cuboid.

Finally, Fig.7, shows sample snapshots from the results obtained on a sequence of a hand performing fine manipulation of an elongated cuboid. Visual inspection confirms that the accuracy of *HOPE* is quite satisfactory, despite the observed complex hand-object interaction.

Sample videos out of these experiments are provided as supplemental material to this submission. More videos with results together with the original benchmark datasets will become available online.

4. Discussion and conclusions

In a hand-object interaction scenario, the observation of hands provides information that is important to understanding the state of the object and vice versa. In this paper, we demonstrated that by considering jointly the hand and the object, it is possible to better understand aspects of both. More specifically, the optimization over the parameters of a joint hand-object 3D model results in full hand pose estimation that is performed more accurately compared to methods that consider the hand in isolation. On top of that, a parametric expression of the manipulated object is also computed. The defined joint optimization problem is solved through Particle Swarm Optimization which proves very competent in handling the complex, multidimensional and multimodal objective function of this problem. From a computational point of view, the proposed approach has only a minor overhead over the case of treating hands in isolation. Results from ex-

tensive experiments on simulated data demonstrated the potential of the method against ground truth, but also comparatively to the results of a state-of-the-art hand pose estimation method that considers hands in isolation. Experiments in real world sequences provide evidence that the proposed method performs well in challenging cases of complex hand articulation and hand-object interaction. Ongoing research considers more flexible 3D models of a hand and also aims at exploiting the fact that a hand and an object cannot share the same physical space (see Sec. 2.3). Additionally, we investigate the potential of *HOPE* to support the understanding of the semantics of human grasping and manipulation activities.

Acknowledgements

This work was partially supported by the IST-FP7-IP-215821 project GRASP.

References

[1] I Albrecht, J Haber, and H.P. Seidel. Construction and animation of anatomically based human hand models. In *2003 ACM SIGGRAPH/Eurographics symposium on Computer Animation*. Eurographics Association, 2003.

[2] A.A. Argyros and M.I.A. Lourakis. Real-time tracking of multiple skin-colored objects with a possibly moving camera. In *ECCV*, 2004.

[3] Vassilis Athitsos and Stan Sclaroff. Estimating 3d hand pose from a cluttered image. *CVPR*, 2:432, 2003.

[4] John Canny. A computational approach to edge detection. *PAMI*, 8(6):679–698, 1986.

[5] M. Clerc and J. Kennedy. The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1):58–73, 2002.

[6] M. de la Gorce, N. Paragios, and D.J. Fleet. Model-based hand tracking with texture, shading and self-occlusions. In *CVPR*, pages 1–8, 2008.

[7] Ali Erol, George Bebis, Mircea Nicolescu, Richard D. Boyle, and Xander Twombly. Vision-based hand pose estimation: A review. *CVIU*, 108(1-2):52–73, Oct. 2007.

[8] Abhinav Gupta, Aniruddha Kembhavi, and Larry S. Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *IEEE Trans. on PAMI*, 31:1775–1789, 2009.

[9] James. Kennedy, R.C. Eberhart, and Yuhui. Shi. *Swarm intelligence*. Morgan Kaufmann Publishers, 2001.

[10] Hedvig Kjellstrom, Javier Romero, David Martinez, and Danica Kragic. Simultaneous visual recognition of manipulation actions and manipulated objects. In *ECCV*, 2008.

[11] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. *CVPR 2010*, 0:2929–2936, 2009.

[12] Thomas B. Moeslund, Adrian Hilton, and Volker Krüger. A survey of advances in vision-based human motion capture and analysis. *CVIU*, 104(2-3):90–126, Dec 2006.

[13] I. Oikonomidis, N. Kyriazis, and A. Argyros. Markerless and efficient 26-dof hand pose recovery. In *ACCV*, 2010.

[14] Aude Oliva and Antonio Torralba. The role of context in object recognition. *Trends in Cognitive Sciences*, 11(12):520 – 527, 2007.

[15] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *ICCV*, Oct 2007.

[16] James M. Rehg and Takeo Kanade. Visual tracking of high dof articulated structures: An application to human hand tracking. In *ECCV*. Springer-Verlag, 1994.

[17] Javier Romero, H. Kjellström, and Danica Kragic. Hands in action: Real-time 3D reconstruction of hands in interaction with objects. In *ICRA*, 2010.

[18] Javier Romero, Hedvig Kjellstrom, and Danica Kragic. Monocular real-time 3D articulated hand pose estimation. *IEEE-RAS Int'l Conf. on Humanoid Robots*, Dec 2009.

[19] Rmer Rosales, Vassilis Athitsos, Leonid Sigal, and Stan Sclaroff. 3d hand pose reconstruction using specialized mappings. In *ICCV*, 2001.

[20] B. Stenger, P.R.S. Mendonca, and R. Cipolla. Model-based 3D tracking of an articulated hand. *CVPR*, pages II–310–II–315, 2001.

[21] EB Sudderth, MI Mandel, WT Freeman, and AS Willsky. Visual hand tracking using nonparametric belief propagation. In *CVPR Workshop*, 2004.

[22] B. White and M. Shaw. Automatically tuning background subtraction parameters using particle swarm optimization. In *IEEE ICME*, 2007.

[23] Ying Wu and Thomas S. Huang. View-independent recognition of hand postures. In *CVPR*, pages 88–94, 2000.

[24] Bangpeng Yao and Li Fei-Fei. Grouplet: A structured image representation for recognizing human and object interactions. In *CVPR 2010*, Jun 2010.

[25] Bangpeng Yao and Li Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. In *CVPR 2010*, Jun 2010.

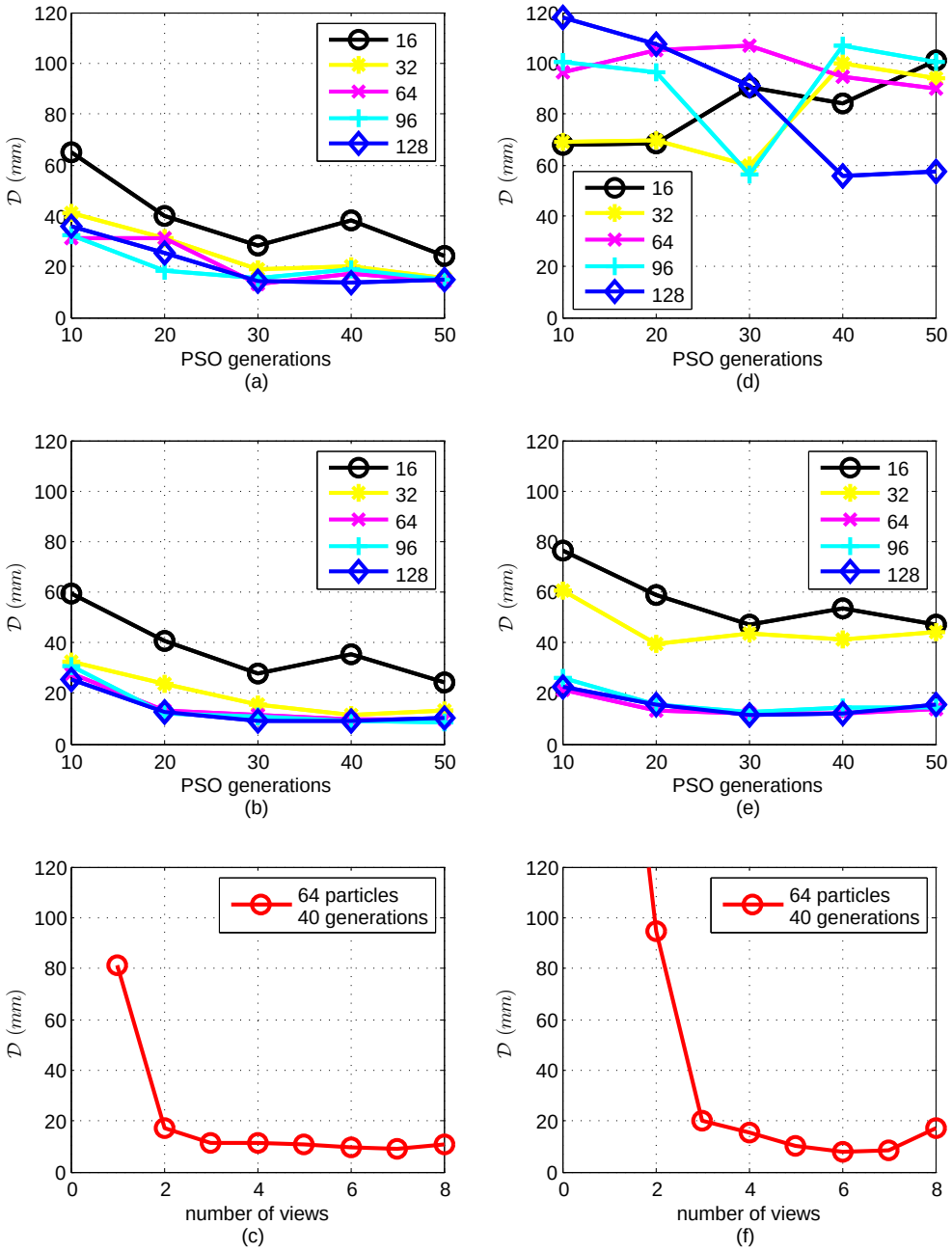


Figure 3: Mean error $\mathcal{D}$ for hand pose estimation (in *mm*) for *HOPE* (left column) and *PEHI* (right column) for different PSO parameters and number of views. (a),(d): Varying PSO particles and generations for 2 views. (b),(e): Same as (a),(d) for 8 views. (c),(f): Increasing number of views, 40 generations, 64 particles/generation.

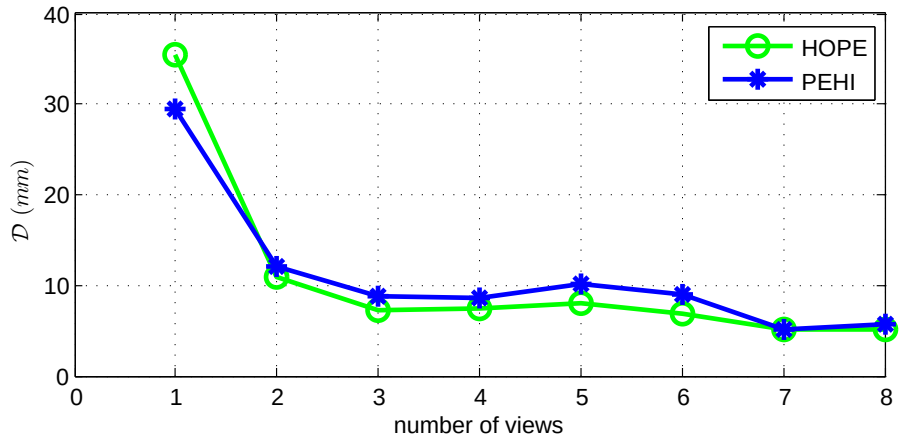


Figure 4: Performance of *HOPE* and *PEHI* on a synthetic sequence of multiframes that shows hands in isolation. For both algorithms, $\mathcal{D}$ is plotted as a function of the number of employed views. 64 PSO particles and 40 generations have been used in both cases.

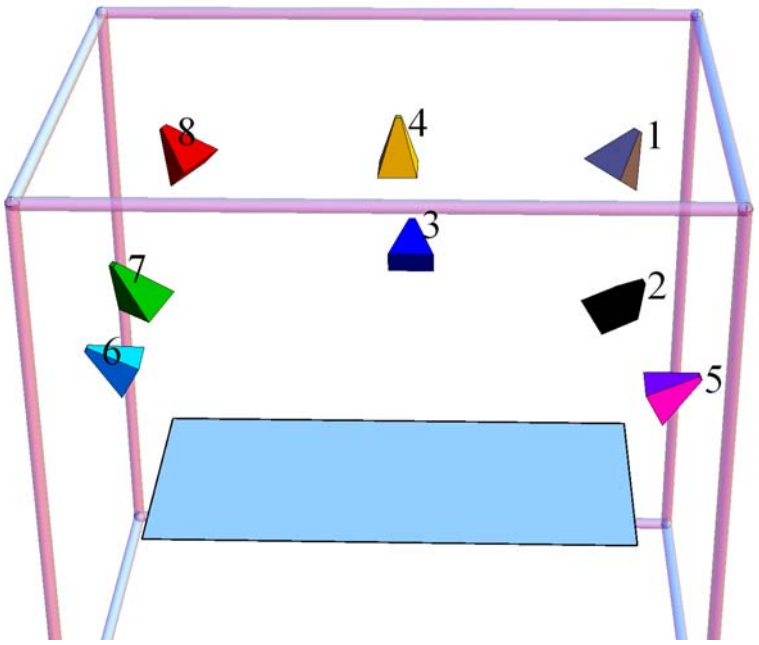
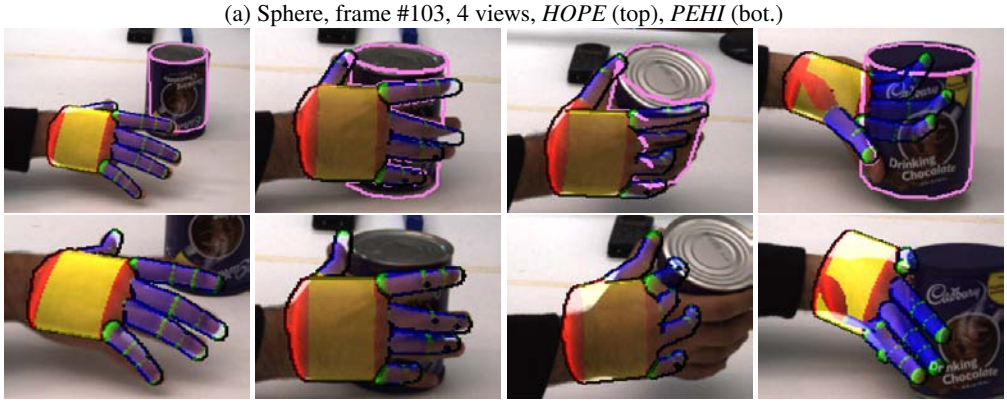
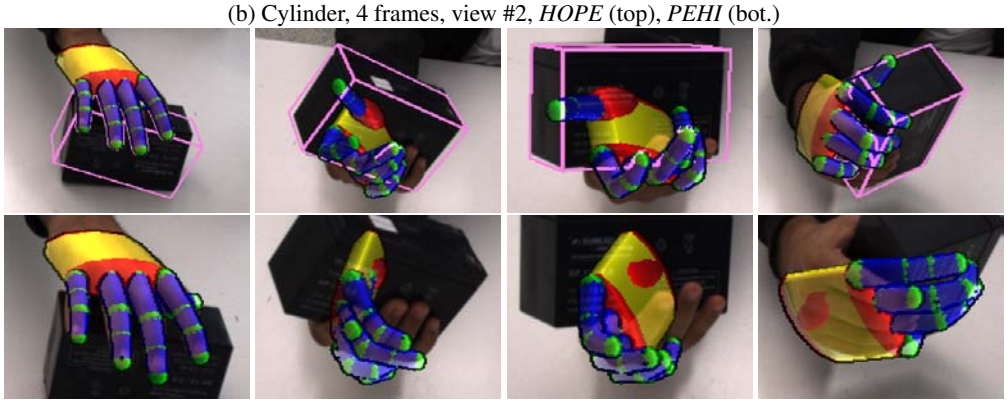


Figure 5: Camera setup for the experiments with real data.

(a) Sphere, frame #103, 4 views, *HOPE* (top), *PEHI* (bot.)



(b) Cylinder, 4 frames, view #2, *HOPE* (top), *PEHI* (bot.)



(c) Box, 4 frames, view #1, *HOPE* (top), *PEHI* (bot.)

Figure 6: Sample frames from the results obtained by *HOPE* and *PEHI* in real-world experiments. For *HOPE* the projection of the estimated 3D object model is shown in pink color.

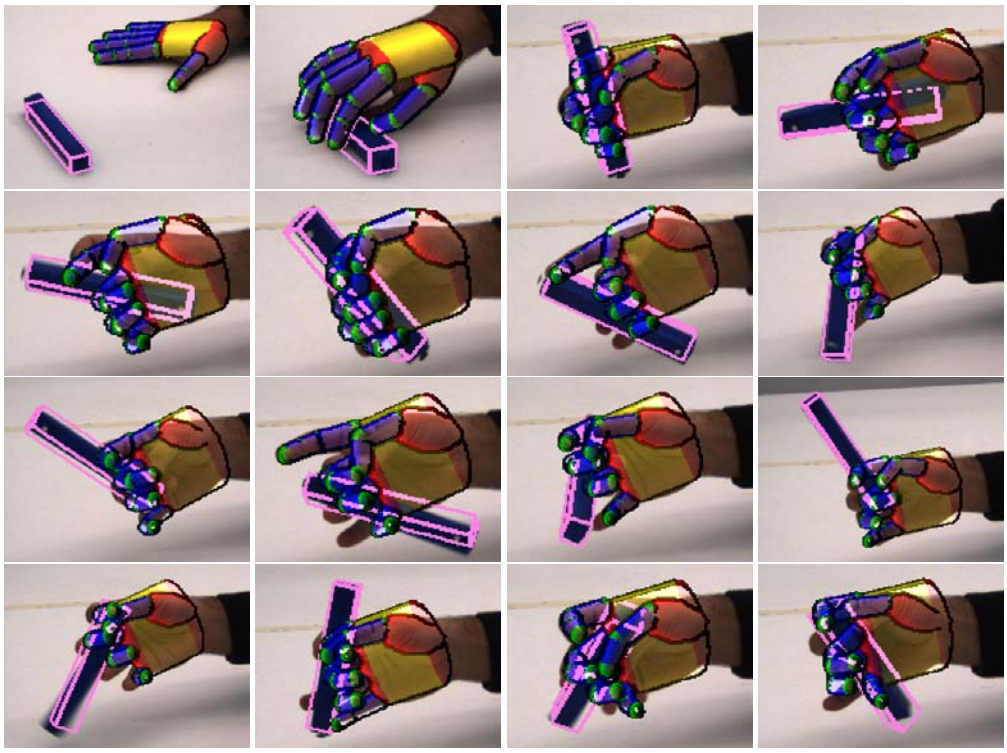


Figure 7: Snapshots from an experiment where a hand performs a complex manipulation of an elongated cuboid.

Particle Filter-Based Fingertip Tracking with Circular Hough Transform Features

Martin Do Tamim Asfour Rüdiger Dillmann

Abstract

In this work, we present a fingertip tracking framework which allows observation of finger movements in task space. By applying a multi-scale edge extraction technique, an edge map is generated in which low contrast edges are preserved while noise is suppressed. Based on circular image features, determined from the map using Hough transform, the fingertips are accurately tracked by combining a particle filter and a subsequent mean-shift procedure. To increase the robustness of the proposed method, dynamical motion models are trained for the prediction of the finger displacements. Experiments were conducted on various image sequences from which statements on the performance of the framework can be derived.

1 Introduction

Towards an intuitive and natural interface to machines, markerless human observation has become a major research focus during the past years. Regarding coarse granular human tracking incorporating the torso, arms, head, and legs, considerable progress has been achieved whereas human hand tracking still remains an unresolved issue, although the hand is considered to be one of the most crucial body parts regarding the interaction with other humans and the environment.

Regarding markerless tracking and detection of human body parts, most systems have limited sensor capabilities which in common case are limited to a stereo camera setup. Full hand tracking approaches in joint angle space have been proposed in [1],[2],[3]. However, due to the highly complex structure of the hand whose motion involves 27 DoF, tracking can be only achieved at a low frame rate or on multiple views from different perspectives.

Using stereo vision, a reasonable solution lies in reducing dimensionality of the problem by shifting from joint angle space into task space. In [4], a finger tracking approach based on Active contours is presented for air-writing. The target to be tracked consists of a contour which is laid around the pointing finger. As a result, since no reliable statement can be made on the actual fingertip position, one has to assume that finger pose is not changing.

Hence, based on curvature properties, in [5] fingertips are detected within a contour which is extracted from skin blob tracking. A more elaborate approach is presented in [6] where particles are propagated from the center of the hand to positions close to the contour. Intersection of the contour with line segments at particles and examination of the transitions between non-skin and skin-area indicate whether a particle represents a fingertip. However, this method is specifically designed to detect tips of stretched fingers. Based on multi-scale color features [7] introduces a hierarchical representation of the hand consisting of blobs of different sizes with each blob representing a part of the hand. The blob features are matched with a number hierarchical 2D models each incorporating a specific finger pose. Therefore, tracking is accomplished under the assumption that the local finger poses regarding the hand remains fixed. In order to implement continuous fingertip tracking method, we would like to rely on prominent features which can be extracted at any time of an image sequence. In [8] for detecting a guitarist's fingertips, circular features are proposed which are localized by performing a circular Hough transform. For the same application, [9] defined semi-circular templates which are used to find the fingers' positions.

In our work, we adopt the concept of circular features to tackle the more complex problem of tracking fingertips of freely moving hand, where overlap of finger and palm occur frequently leading to difficulties regarding the robust extraction of these features. For tracking, we combined particle filtering with a mean-shift algorithm. In addition, a dynamical motion model for predicting was trained to enhance the robustness of the proposed framework.

The paper is organized as follows. Section 2 describes the feature extraction consisting of a edge detection step and the Hough transform. Details on the tracking procedure performed on the resulting map are given in Section 3. Subsequently, first experiments with the framework are explained in Section 4. In Section 5, the work is summarized and notes to future works are given.

2 Feature Extraction

In order to generate the edge image, a skin color segmentation is performed for extracting the hand and finger regions. Morphological operators are applied on the segmented image to eliminate noise and to produce a uniform region. To detect the edges in this preprocessed image, image gradients are calculated on various scales.

2.1 Multi-Scale Edge Extraction

Considering the problem of fingertip tracking, due to small intensity variances between different parts of the hand, e.g. the fingernail and the skin, respectively, the finger regions and the palm, it is desired to detect edges where contrast can vary over a broad range. Depending on the parameters, applying standard algorithms, such as the Canny edge detectors on a wider scale, leads to an edge image where numerous, false edges occur. To preserve low contrast edges in certain areas while reducing noise close to high-contrast edges, based on the work of [10], we implemented a filter approach consisting of a steerable Gaussian derivative filter on multiple

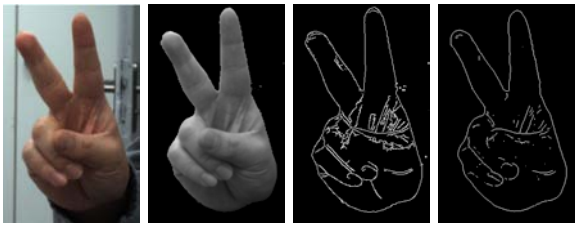


Figure 1. Left: Original input image. Center/Left: Color segmented image. Center/Right: Edge image using Canny detector. Right: Edge image using the method proposed in Section 2.1.

scales. The basis filter for x is defined as follows:

$$G_k^x(x,y;\sigma_k) = \frac{-x}{2\pi\sigma_k^4} e^{\frac{-(x^2+y^2)}{2\sigma_k^2}}. \quad (1)$$

$G_k^y(x,y;\sigma_k)$ is defined analogously. To determine the scale at which a gradient can be reliably estimated, the magnitude of the filter response $r_k^x(x,y;\sigma_k)$ and $r_k^y(x,y;\sigma_k)$, obtained by convolution of the image I with the filters from Eq. 1, is checked against a noise threshold. While the magnitude can be calculated according to:

$$r_k^m(x,y;\sigma_k) = \sqrt{r_k^x(x,y;\sigma_k)^2 + r_k^y(x,y;\sigma_k)^2}, \quad (2)$$

the threshold is set by following function:

$$c_k = \frac{\sqrt{-2ln(1-(1-\alpha)^R)}}{2\sigma_k^2\sqrt{2\pi}} s_l. \quad (3)$$

with s_l representing the standard deviation and α the significance level for an image with R pixels which defines an upper boundary for allowed misclassification of image pixels. To take into account local intensity and contrast conditions, we focus on local signal noise in a specific region rather than on global sensor noise. Therefore, Eq. 3 depends on the local standard deviation s_l calculated within a $2\sigma_k^{max} \times 2\sigma_k^{max}$-neighborhood where σ_k^{max} denotes the largest scale being examined. Hence, we calculate each gradient at the minimum reliable scale σ_k^{min} where the likelihood of error due local signal noise falls below a standard tolerance. This guarantees that a more accurate gradient map is estimated which is less sensitive to signal noise and errors caused by interference from nearby structures. The edge image obtained from the map is depicted in Fig. 1.

2.2 Hough Transformation for Circle Detection

The circle features representing the N fingertips are detected by applying a Hough transform with radius r. For each edge point (x,y) with known direction in the form of a rotation angle θ, a vote is assigned to possible circle feature positions (u,v) in two-dimensional Hough space I_H according to:

$$I_H(u,v) = I_H(u,v) + 1 \quad (4)$$

with $u = x \pm rcos(\theta)$ and $v = y \pm rsin(\theta)$. Unfortunately, curves around the fingertips do not always feature perfect circular arcs. To cope with noisy and slightly deformed curves, the voting is performed for a set of radii $R = \{-m1.1r, \ldots, m1.1r\}$ with $m \in \mathbb{N}$ whereby a range of pixels along the edge tangent is considered during the voting process. In order to increase the robustness of the tracking algorithm, a density distribution is formed in Hough space by convolving I_H with a Gaussian kernel $G(u,v;\frac{r}{2})$.

Since the hand motion occurs in 3D Euclidean space, a fixation of r is only valid if movement of the fingertip in direction of the z-axis of a camera is excluded during the tracking. Adaptation of r in each frame, allows to track fingers in all directions. Based on the generated density distribution in frame t a radius estimate $\hat{r}_t$ is determined by applying an Expectation Maximization algorithm. Further details are given in Section 3.3.

3 Tracking Fingertips

3.1 Prediction

Providing a prediction on the movement of the objects to be tracked increases the robustness of a statistical tracking framework. We train dynamical motion models in the form of a second-order auto-regresssive (AR) process as proposed in [4], which is described as follows:

$$q_t - \bar{q} = A_1(q_{t-1} - \bar{q}) + A_2(q_{t-2} - \bar{q}) + b_0\omega_k \quad (5)$$

where $q_t \in \mathbb{R}^D$ denotes the current configuration, $\bar{q}$ the mean configuration, and $\omega_k \in [0,1]$. To learn the AR parameters A_1, $A_2 \in \mathbb{R}^{D\times D}$ and $b_0 \in \mathbb{R}^D$, training data is provided in the form of a configuration sequence $Q = \{q'_0, \ldots, q'_M\}$ whereas the sequence is generated by manual labeling of fingertips in each frame of a recorded image sequence.

Two AR models are trained to provide predictions for the local fingertip movement concerning a static hand pose as well as the movement of the hand itself. Based on the assumption that the motion of each finger is influenced by the motion of the neighbored fingers, the first model is trained with training data whose instances $q'_i \in Q$ with $D = N$ consists of the length of $v_t^{j,j+1} = p_t^{j+1 \bmod N} - p_t^j$ between the fingertips j and $j+1 \bmod N$:

$$q'_i(j) = \|v_t^{j,j+1}\| \;\; j = 1, \ldots, N. \quad (6)$$

For finger j, this leads to following displacement vector:

$$\bar{v}_t(j) = \frac{q_t^j - q_{t-1}^j}{q_{t-1}^j} v_t^{j,j+1} - \frac{q_t^{j-1} - q_{t-1}^{j-1}}{q_{t-1}^{j-1}} v_t^{j-1,j}. \quad (7)$$

The second model which considers the global movement of the average position of all fingertips p_{mean} is trained with a data set formed of $q'_i = p_{mean}$ with $D = 2$ resulting into a overall displacement:

$$\hat{v}_t(j) = q_t^j + \bar{v}_t(j). \quad (8)$$

Due to the coupled fingertip movements, the models behave well resulting in reasonable prediction of the finger displacements which supports the state estimation in the ensuing tracking procedure.

3.2 Particle Filter Tracking

For the proposed fingertip tracking framework, a state hypothesis s of a particle (s,w) consists of the

N fingertip positions of the hand with each position being denoted by the coordinates (x,y) within the image. Particle filtering is an iterative algorithm where, first, at time t samples are drawn from a set of previous particles $X_{t-1} = \{(s^i_{t-1}, w^i_{t-1})\}$ proportionally to their likelihood w^i_{t-1}. Subsequently, from each drawn sample a new state hypothesis s^i_t is generated. Adding a Gaussian random variable ω and the displacement vector $\hat{v}_j$ from Eq. 8, s^i_t can be written as:

$$s^i_t = s^i_{t-1} + \hat{v}_t + \omega. \tag{9}$$

To determine a particle set X_t, for each s^i_t the likelihood w^i_t is computed. In order to compute the weights for the new set of particles, one has to approximate the likelihood function $p(z_t|s_t)$ with z_t representing the current observation. Our approximation of $p(z_t|s_t)$ is based on two cues: a contour and a distance cue. The contour cue is derived by exploiting the external energy functional E_{img} of a contour C^i_t obtained by connecting the single points in s^i_t according the finger order. The E_{img} is determined in terms of an edge image Z^E_t which is constructed by drawing lines between the a set of maximum bins Z^V_t that can be found in I_H. As a result, the likelihood function can be written as:

$$p_c(z_t|s_t) \propto w_c(s_t) = exp\left\{\frac{-E_{img}(Z^E_t, C_t)}{\sigma^2_c}\right\}. \tag{10}$$

The distance cue is calculated from the Euclidean distance between s^i_t and Z^V_t which consists of the sum of minimal distances between $s^i_t(j)$ and Z^V_t. Based on this cue, the likelihood function can be defined as

$$p_d(z_t|s_t) \propto w_d(s_t) = exp\left\{\frac{-\sum^N_{j=1} \min(\|s_t(j) - Z^V_t\|)}{N\sigma^2_d}\right\}. \tag{11}$$

The final likelihood function is constructed from Eq. 10 and Eq. 11, hence, we define the computation of weights as follows:

$$w^i_t = \frac{\sqrt{w_c(s^i_t) w_d(s^i_t)}}{\sum^M_{k=1} \sqrt{w_c(s^k_t) w_d(s^k_t)}}. \tag{12}$$

One obtains a current state estimate of the fingertip configuration by evaluating the following sum:

$$s_t = \sum^M_{i=1} w^i_t s^i_t. \tag{13}$$

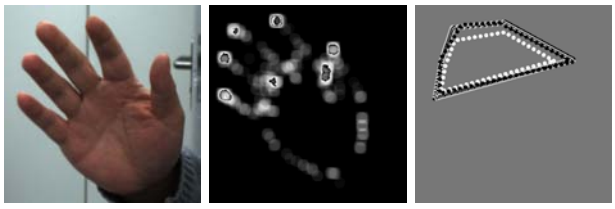


Figure 2. Left: Original input image. Center: Visualization of the Hough space. Right: Generated contour for the particle filter tracking. The particle with the highest likelihood (black dotted line) and the particle with the lowest (white dotted line) are depicted.

Further details concerning the particle filter algorithm can be found in [11].

3.3 Mean-Shift

To obtain more accurate position estimates, a mean-shift algorithm is applied to move the estimated fingertip position $p_n = s_t(n)$ towards the peak of local density distribution. We adopted the EM-like mean-shift algorithm proposed in [12] which in addition provides the possibility to estimate the covariance of the local density distribution. The covariance estimation allows us to adapt the radius r corresponding to the current circular image features. Hence, taking into account movement in the depth of the camera, for tracking circular features in Hough space one has to incorporate an adaptation of radius r_t. Under the assumption that the distribution can be modeled as a Gaussian, we want to find parameters $\hat{p}_n$ and V_d representing center and covariance matrix of the distribution that maximize following function:

$$f(\hat{p}_n, V_d) = \sum^M_{j=1} G(p_j; \hat{p}_n; V_d) I_H(p_j). \tag{14}$$

which can be solved iteratively by, first, calculating λ_j according to:

$$\lambda_j = \frac{G(p_j; \hat{p}_n; V_d)) I_H(p_j)}{\sum^M_{j=1} G(p_j; \hat{p}_n; V_d) I_H(p_j)} \tag{15}$$

and then determining a new estimation for center which can be written as:

$$\hat{p}^{i+1}_n = \sum^M_{j=1} \lambda_j p_j \tag{16}$$

whereas a covariance matrix estimation is obtained by evaluating following term:

$$V^{i+1}_n = c \sum^M_{j=1} \lambda_j (p_j - \hat{p}^i_n)(p_j - \hat{p}^i_n)^T \tag{17}$$

whereas c is a constant. If convergence is achieved, the radius is determined from the covariance matrix.

4 Results

The proposed fingertip tracking framework was applied on several image sequences which were captured with a static stereo camera setup and a resolution of $R = 640 \times 480$ pixels. For edge extraction, the method presented in Section 2.1 is applied with $\sigma_k = 4, 2, 1, 0.5$ and $\alpha = 0.5$. Currently, initialization of the tracking is done manually by defining a region I^n_H where finger n is to be found. Using the Hough transform with different radii constructed with $m = 3$, The maximum bin in I^i_H is labeled as finger n according to the finger order $n = \{Thumb = 0, Index = 1, Middle = 2, Ring = 3, Pinkie = 4\}$.

Taking into account the predicted displacements of the fingers, the particle filter tracking algorithm with minimum 600 particles shows good performance. Around 3 mean-shift iterations needed to achieve convergence, The number of iterations for the subsequent mean-shift algorithm depends on the numbers of particles meaning less particles more mean-shift iterations

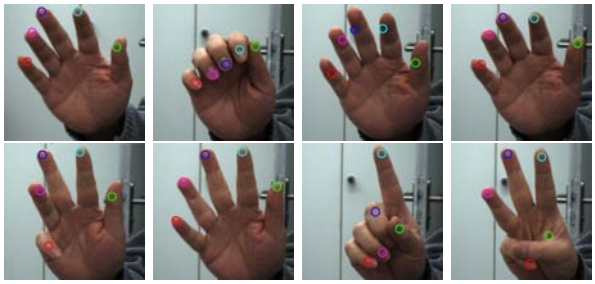


Figure 3. Images of the tracking results. The upper row depicts simultaneous closing of the fingers, while the lower row shows sequential flexing of the fingers. The fingertips are labeled as follows: Thumb (green), index (light blue), middle (dark blue), ring (pink), and pinkie (red).

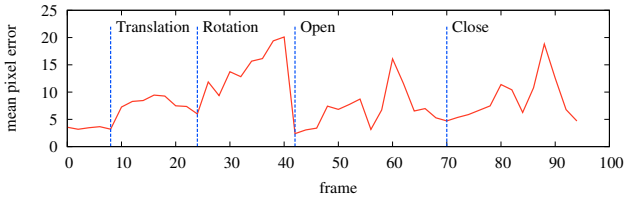


Figure 4. Error plot for a sequence of four hand and finger movements: Translation of the hand, rotation of the hand, close and open movement incorporating flexing the fingers.

vice versa. Exploiting the combination of both, reasonable accuracy of ≈ 7 pixels mean deviation is achieved for translation movements. For a rotation, opening and closing movement, the error increases up to 20 pixels. These measurements are depicted in Fig. 4. The tracking procedure fails if a finger is lost, which is the case if the movements are too fast.

In case of failure, currently, a very rudimentary reinitialization is performed consisting of a search for maximum bins in the vicinity of the last known estimation and arranging of the fingers according accord position polar space, assuming that fingers are arranged clockwise, respectively counter clockwise, depending on the hand that is observed. Since this assumption is not valid for several finger poses, hence, this might lead to mislabeling.

Since this algorithm operates on monocular images, for each view a tracking instance is created whereas the 3D positions of the fingers are calculated by exploiting epipolar geometry. The presented framework is capable of online tracking of fingertip motion with a frame rate of 15 Hz on a 2.40 GHz dual core CPU. Sample images during the tracking process are depicted in Fig. 3.

5 Conclusion

In this work, we have presented a fingertip tracking which allows observation of fine granular human actions such as grasping in an efficient manner. Using Hough transform and a combination of particle filter and mean-shift tracking, circular features representing the fingertips could be localized and tracked. Currently, the proposed framework is applied for capturing human grasping movements for online imitation learning using the on-board systems, a pair of stereo cameras, of a robot.

However, in the experiments we conducted, we were able to observe that the error on the localization of the fingertips increases, when the hand performs movements which go beyond translation. These can be led back to the use of a single dynamical motion model for the prediction. In the near future, the local fingertip prediction module will be implemented in the form of a net of multiple intertwined motion models in order to provide better predictions. Concerning the motion model of the hand, we realized that it needs to be extended by an angular dimension to cover the rotation of the hand. To enable full online observation of the human upper body the fingertip tracking will be integrated into a upper body tracker and its implementation will be improved to raise its efficiency.

References

[1] J. Rehg and T. Kanade, "Digiteyes: Vision-based hand tracking for human-computer interaction," in *Proc. Workshop Motion of Non-Rigid and Articulated Bodies*, November 1994, pp. 16–22.

[2] B. Stenger, P. R. S. Mendonca, and R. Cipolla, "Model-based 3d tracking of an articulated hand," in *Proc. Int. Conf. Computer Vision and Pattern Recognition*, December 2001, pp. 310–315.

[3] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, "Markerless and efficient 26-dof hand pose recovery," in *Proc. 10th Asian Conf. Computer Vision*, November 2010.

[4] A. Blake and M. Isard, *Active Contours: The Application of Techniques from Graphics, Vision, Control Theory and Statistics to Visual Tracking of Shapes in Motion*, 2000.

[5] A. A. Argyros and M. Lourakis, "Vision-based interpretation of hand gestures for remote control of a computer mouse," in *Proc. HCI06 Workshop*, May 2006, pp. 40–51.

[6] K. J. Hsiao, T. W. Chen, and S. Y. Chien, "Fast fingertip positioning by combining particle filtering with particle random diffusion," in *Proc. Int. Conf. Multimedia and Expo*, June 2008, pp. 977–980.

[7] L. Bretzner, I. Laptev, and T. Lindeberg, "Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering," in *Proc. Int. Conf. Aut. Face and Gesture Recognition*, May 2002, pp. 423–428.

[8] A. M. Burns and M. M. Wanderley, "Visual methods for the retrieval of guitarist fingering," in *Proc. Int Conf. New Interfaces for Musical Expression*, Paris, France, June 2006, pp. 196–199.

[9] C. Kerdvibulvech and H. Saito, "Markerless guitarist fingertip detection using a bayesian classifier and a template matching for supporting guitarists," in *Proc. 10th Virtual Reality Int. Conf.*, April 2008.

[10] J. Elder and S. Zucker, "Scale space localization, blur, and contour-based image coding," in *Proc. Int. Conf. Computer Vision and Pattern Recognition*, June 1996.

[11] P. Azad, *Visual Perception for Manipulation and Imitation in Humanoid Robots*, 2009.

[12] Z. Zivkovic and B. Kroese, "An em-like algorithm for color-histogram-based object tracking," in *Proc. Int Conf. Computer Vision and Pattern Recognition*, Washington D.C., USA, June 2004, pp. 798–803.

Computer Vision and Image Understanding 115 (2011) 105–116

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Unsupervised learning of background modeling parameters in multicamera systems

Konstantinos Tzevanidis [a,b], Antonis Argyros [a,b,*]

[a] Computer Science Department, University of Crete, Knossou Ave., GR 71409 Heraklion, Crete, Greece
[b] Institute of Computer Science, FORTH, N. Plastira 100, Vassilika Vouton, GR 70013, Heraklion, Crete, Greece

ARTICLE INFO

Article history:
Received 16 March 2010
Accepted 19 September 2010
Available online 25 October 2010

Keywords:
Background modeling
Foreground detection
Multicamera consensus
Particle Swarm Optimization
Camera networks

ABSTRACT

Background modeling algorithms are commonly used in camera setups for foreground object detection. Typically, these algorithms need adjustment of their parameters towards achieving optimal performance in different scenarios and/or lighting conditions. This is a tedious process requiring considerable effort by expert users. In this work we propose a novel, fully automatic method for the tuning of foreground detection parameters in calibrated multicamera systems. The proposed method requires neither user intervention nor ground truth data. Given a set of such parameters, we define a fitness function based on the consensus built from the multicamera setup regarding whether points belong to the scene foreground or background. The maximization of this fitness function through Particle Swarm Optimization leads to the adjustment of the foreground detection parameters. Extensive experimental results confirm the effectiveness of the adopted approach.

1. Introduction

As digital cameras become cheaper, multicamera setups or camera networks are becoming commonplace. Calibrated multiview setups are associated with some strong assumptions and their intrinsic/extrinsic calibration is a tedious process. Nevertheless, their ability to reduce occlusion effects and appearance ambiguities leads to more robust performance of computer vision algorithms, a fact that typically outweighs their disadvantages. Several multicamera-based applications such as semi-automated surveillance [8], target tracking [17], 3D video recording [18,23], human motion modeling [4,28] and sports analysis [9] perform object detection, most commonly using some background modeling-based foreground detection method. Thus, such methods constitute important ingredients of modern multiview computer vision systems.

A common drawback of several existing foreground detection methods is that their performance critically depends on several parameters that require considerable expertise in order to be adjusted properly. Unfortunately, there is no universal parameter set that can generalize optimally across the different conditions that may be encountered. In the typical case, different scenarios that exhibit variable degree of occlusions (e.g., crowded scenes), stopped targets, clutter motion (e.g., flowing water) and global or local illumination changes, require different tuning of the algorithm towards high quality results. Despite its great importance, proper parameter tuning is often overlooked resulting in suboptimal foreground detection output. The need for adaptive parameter tuning is even more pronounced when dealing with online, real-time applications that capture endless video streams (e.g., automated surveillance) where the environmental and other conditions might change considerably over time.

One of the few approaches that deal with this problem is the one adopted by White and Shaw [27], which presents a method that optimizes background subtraction with respect to given ground truth. More specifically, the goal is to optimize two basic parameters of a background subtraction algorithm [24] that is applied to an image sequence acquired by a single camera. The required ground truth consists of manually defined foreground silhouettes. The *F* measure [22] between the silhouettes calculated by the background subtraction algorithm and the ground truth silhouettes constitutes the fitness function of a given parameter set. Finally, *Particle Swarm Optimization* (PSO) is employed to maximize this fitness function by searching over the space of possible background subtraction parameters.

In this work, we propose a novel method for automatically tuning the foreground detection parameters, utilizing information taken by a multicamera setup. In contrast to White and Shaw [27], the proposed method does not require user intervention at any point of the process and does not assume the availability of ground truth measurements. Thus, it can be applied to the automatic tuning of foreground detection performed on any system that captures endless video streams where ground truth information is not available. Similar to White and Shaw [27], we employ PSO to optimize a fitness function that is defined over a multidimensional foreground detection parameter space.

* Corresponding author. Address: N. Plastira 100, Vassilika Vouton, GR-700-13 Heraklion, Crete, Greece. Fax: +30 2810 391609.
E-mail address: argyros@ics.forth.gr (A. Argyros).

1077-3142/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2010.09.003

Instead of using ground truth silhouette images, we employ *confidence maps* that are calculated through the fusion of the foreground images estimated by the multicamera setup. At each step, one such map is produced for every camera of the configuration. Each confidence map consists of scores that represent the cumulative confidence in the multicamera setup regarding whether a pixel belongs to the foreground or not. For each and every camera, the fitness function measures the similarity of the foreground estimate to the confidence map. The fundamental idea behind the definition of the fitness function is that if several cameras agree that a certain point in the scene belongs to the foreground, then this is likely to be so. False positives and false negatives may exist in the process. Nevertheless, it is very unlikely that a consensus will be build around them. As in [27], PSO is used to maximize the fitness function. PSO suggests foreground detection parameters that produce new confidence maps which, in turn, suggest new parameters. The termination of this iterative process provides the parameter vector found to achieve the greatest fitness. Through a series of experiments, we show that both the defined fitness function and optimization process are very suitable for effectively solving the problem of unsupervised adjustment of foreground detection parameters.

The main contributions of this work are (1) the definition of multicamera consensus and the resulting confidence maps in the optimization of the foreground detection parameters, (2) the unsupervised solution of the problem of parameter tuning as opposed to the previous supervised methods requiring ground truth information, and (3) a thorough experimental study of the behavior of the proposed approach with a detailed investigation of various factors that may affect its performance.

The remainder of this paper is organized as follows. In Section 2 the foreground detection algorithm that is used throughout this work is presented. It has to be noted that the selection of the particular method is based on its popularity and performance [3]. Nevertheless, the proposed method can, in principle, be applied to any other background subtraction/foreground detection method. Section 3 defines the confidence maps that guide the optimization process. Section 4 presents the employed optimization algorithm. Section 5 provides a detailed description of the proposed algorithm. Experiments and results are presented in Section 6. Finally, a brief summary and conclusions is given in Section 7.

2. Background modeling and foreground detection

Background modeling and foreground detection is a way to detect moving objects in views acquired by static cameras. The great importance of such methods has given rise to several approaches. According to Piccardi [21], such methods typically operate at the pixel level. The simplest ones directly subtract the average, median or running average of a number of frames from the current view. Other methods use kernel density estimators and mean-shift based estimation [10,12]. In [20], the notion of eigen-background is defined.

One of the best performing methods is the one proposed by Stauffer and Grimson [24] that models the appearance of each image pixel as a mixture of Gaussians. Because of its effectiveness and popularity [3], our work considers this method as the basis of the proposed, unsupervised parameter optimization approach. More specifically, we employ the variant proposed by Zivkovic [29]. For the sake of self completeness, an introduction to this method is provided.

Given a sequence of images, let $\vec{x}^{(t)}$ be a pixel of image $I^{(t)}$ at time t in some colorspace (i.e., RGB). The background model is estimated from a training set $X_T = \{x^{(t)}, \ldots, x^{(t-T)}\}$ where T determines the time period for which the model's history is extended. Each pixel is modeled as a M component Gaussian Mixture Model (GMM) given by

$$\hat{p}(\vec{x}|X_T, fb) = \sum_{m=1}^{M} \hat{\pi}_m N(\vec{x}; \hat{\vec{\mu}}_m, \hat{\sigma}_m^2 I), \tag{1}$$

where $\hat{\vec{\mu}}_1, \ldots, \hat{\vec{\mu}}_M$ are the estimates of the means and $\hat{\sigma}_1, \ldots, \hat{\sigma}_M$ are the estimates of the variances of the GMM components. *fb* denotes the fact that the recent history contains observed values belonging to both the foreground (f) and the background (b). Given a new data sample $\vec{x}^{(t)}$ at time t, the recursive update equations of mixing weights, means and variances are:

$$\hat{\pi}_m \leftarrow \hat{\pi}_m + \alpha\left(o_m^{(t)} - \hat{\pi}_m\right) - \alpha c_T, \tag{2}$$

$$\hat{\vec{\mu}}_m \leftarrow \hat{\vec{\mu}}_m + o_m^{(t)}(\alpha/\hat{\pi}_m)\vec{\delta}_m, \tag{3}$$

$$\hat{\sigma}_m^2 \leftarrow \hat{\sigma}_m^2 + o_m^{(t)}(\alpha/\hat{\pi}_m)\left(\vec{\delta}_m^T\vec{\delta}_m - \hat{\sigma}_m^2\right), \tag{4}$$

where $\vec{\delta}_m = \vec{x}^{(t)} - \hat{\vec{\mu}}_m$, α is the constant that represents an exponentially decaying envelope utilized to attenuate the effect of past data and c_T a small bias factor, typically set to 0.01 (see Zivkovic [29] for details). A sample is close to a GMM component if its Mahalanobis distance from the mode is smaller than a certain threshold, typically set equal to three standard deviations. Based on this, the ownership $o_m^{(t)}$ for a newly arrived sample is set to 1 for the GMM component with the larger mixing weight among all the components that their distances from the sample is less than the predefined threshold and 0, otherwise. The squared distance from the mth component is computed by $D_m^2(\vec{x}^{(t)}) = \vec{\delta}_m^T\vec{\delta}_m/\hat{\sigma}_m^2$. Updates of π_ms must be followed by a normalization so that they add up to one.

Background modeling starts with one GMM component centered on the first sample. While the new samples that arrive are not within three standard deviations from the existing modes of the GMM, new components are generated with $\hat{\pi}_{M+1} = \alpha$, $\hat{\vec{\mu}}_{M+1} = \vec{x}^{(t)}$ and $\hat{\sigma}_{M+1} = \sigma_0$ where σ_0 is the initial variance. During updates, if a mixing weight $\widehat{\pi_m}$ becomes negative, the corresponding mixture component is removed from the GMM and the mixing weights of the remaining components are normalized to sum to one. Moreover, if a newly imported component forces the total number of components to increase beyond a certain threshold, the component with the smallest mixing weight assigned to it is excluded from the mixture.

Given that the components of the mixture are sorted in a descending order of their mixing weights, it is assumed that the background can be modeled by the set B of the largest GMM components as:

$$p(\vec{x}|X_T, fb) \sim \sum_{m=1}^{B} \hat{\pi}_m N\left(\vec{x}; \hat{\vec{\mu}}_m, \sigma_m^2 I\right), \tag{5}$$

where

$$B = \arg\min_j \left\{ j \Big| \sum_{m=1}^{j} \hat{\pi}_m > (1 - c_f) \right\} \tag{6}$$

and c_f is the maximum allowable sum of mixing weights of the GMM components modeling the foreground.

Following the above analysis, an observed pixel is part of the background, if it is found to be close to one of these B Gaussian components. Otherwise, this pixel is assigned the foreground label. An example outcome of this foreground detection method is shown in Fig. 1.

3. Multiview camera setup

The foreground detection method presented in Section 2 operates on an image sequence acquired by a single, static camera.

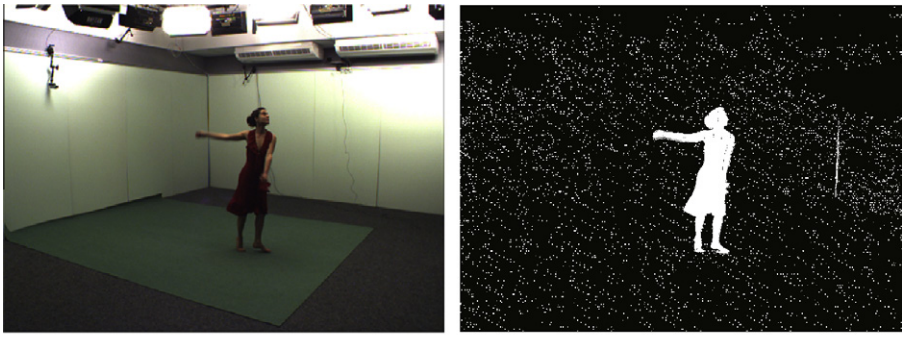


Fig. 1. (a) A frame of Inria's Dancer sequence and (b) foreground detection output. White and black pixels correspond to foreground and background, respectively.

The straightforward approach to performing foreground detection in a multicamera setup is to employ it independently in each of the acquired views. A basic idea behind this work is that the joint observation of a given 3D space by a set of cameras can be used to provide information that may guide the joint optimization of the foreground detection parameters. A given observed 3D point, either belongs to the scene foreground or scene background. Thus, the *visual hull* [15] estimated through *volume intersection* [16] can be used to compute the *multiview configuration consensus* regarding the foreground of a given scene.

More specifically, each camera of the configuration votes in a common voxel space for occupied voxels by projecting an estimation of its own foreground image on this space. The voxel space describes a discretization of the actual space. A voxel can be considered, by a single view, as being occupied by some object or not. This occupancy information is all that is required to calculate the multicamera consensus regarding the objects present in the scene. After the occupancies are calculated, the voxel space can be back-projected to every view to calculate a set of *confidence maps*, one per view. The use of voxel occupancies as a way to combine information from multiple views has been proposed at [11] where a probabilistic framework for fusing silhouette cues is presented.

What follows, is a detailed presentation of how the multicamera consensus and the individual confidence maps are built.

3.1. Multicamera consensus

To calculate the multicamera consensus, a 3D voxel space of the actual scene is defined. This space is sampled to create a 3D grid, $G = \{G^0, G^1, \ldots, G^n\}$ where each $G^c = (X_c, Y_c, Z_c)$ is a 3D point. General perspective projection of a 3D point $(X_c, Y_c, Z_c, 1)$ to a 2D point (x_c, y_c, f_c) on the ith view plane can be calculated given the corresponding projection matrix $P_i = C_i[R_i|T_i]$ through

$$(x_c, y_c, f_c)^T = C_i[R_i|T_i](X_c, Y_c, Z_c, 1)^T, \tag{7}$$

where C_i is the camera calibration matrix, R_i the rotation matrix and T_i the translation vector with respect to a world-centered coordinate system. In the general case, the cameras of a multiview configuration cannot be fully aligned on a common field of view (FOV), so a number of 3D points will fall outside the FOV of some cameras. For the view plane of camera i with dimensions $w_i \times h_i$ we define the function $L_i(x,y)$ that labels the projections falling inside the camera FOV as

$$L_i(x,y) = \begin{cases} 1 & 1 \leqslant x \leqslant w_i \wedge 1 \leqslant y \leqslant h_i, \\ 0 & \text{otherwise}. \end{cases} \tag{8}$$

Furthermore, we denote by S_i the silhouette image (as the one shown in Fig. 1b) taken from camera i, where $S_i(x,y) = 1$ for foreground pixels and $S_i(x,y) = 0$ for background pixels. Occupancy scores $O(X_k, Y_k, Z_k)$ of 3D points of G are computed as

$$O(X_k, Y_k, Z_k) = \begin{cases} 1 & s = l > \frac{|C|}{2} \\ 0 & \text{otherwise} \end{cases}, \quad \forall k \in [0, n]. \tag{9}$$

In Eq. (9), $|C|$ is the number of cameras used. l is termed the *visibility factor* (see Fig. 2a) and s the *intersection factor* (see Fig. 2b). These factors are defined as

$$l = \sum_{i \in C} L_i\left(\frac{x_k^i}{f_k^i}, \frac{y_k^i}{f_k^i}\right), \quad s = \sum_{i \in C} S_i\left(\frac{x_k^i}{f_k^i}, \frac{y_k^i}{f_k^i}\right), \tag{10}$$

where $(x_k^i/f_k^i, y_k^i/f_k^i)$ are the projections of (X_k, Y_k, Z_k) at view plane i.

3.2. Confidence maps

Confidence maps $\mathscr{C}_i(x,y)$ are computed for every view i by accumulating the occupancy scores of the back-projections of the view planes on every slice of the grid G. Slices are considered to be 3D point sets of fixed Z_c, with Z_c taking discrete values in the range of $[Z_{min}, Z_{max}]$. Therefore, confidence maps are calculated through:

$$\mathscr{C}_i(x,y) = \sum_{Z_{min} \leqslant z \leqslant Z_{max}} O(X', Y', z) \tag{11}$$

for every (x,y) such that $1 \leqslant x \leqslant w_i \wedge 1 \leqslant y \leqslant h_i$. Given the 3×4 projection matrix $P_i = [p_{mn}^i]$ of view i the projections X' and Y' of x and y are calculated analytically as

$$Y' = \frac{z(xp_{23}^i - yp_{13}^i + m(yp_{33}^i - p_{23}^i))}{yp_{12}^i - xp_{22}^i - m(yp_{32}^i - p_{22}^i)} + \frac{m(yp_{34}^i - p_{24}^i) + xp_{24}^i - yp_{14}^i}{yp_{12}^i - xp_{22}^i - m(yp_{32}^i - p_{22}^i)} \tag{12}$$

and

$$X' = \frac{Y'(yp_{32}^i - p_{22}^i) + z(yp_{33}^i - p_{23}^i) + yp_{34}^i - p_{24}^i}{p_{21}^i - yp_{31}^i}, \tag{13}$$

where

$$m = \frac{xp_{21}^i - yp_{11}^i}{p_{21}^i - yp_{31}^i}. \tag{14}$$

After their calculation, the values of the confidence maps are normalized to the range [0,1]. The closer a value is to 1, the higher the estimated confidence that the corresponding pixel belongs to a foreground object.

Fig. 3 shows examples of computed confidence maps. As can be verified, confidence maps attenuate the holes in the silhouettes but also the noise in the background. The intuition behind this result is that although false positives and false negatives may exist in indi-

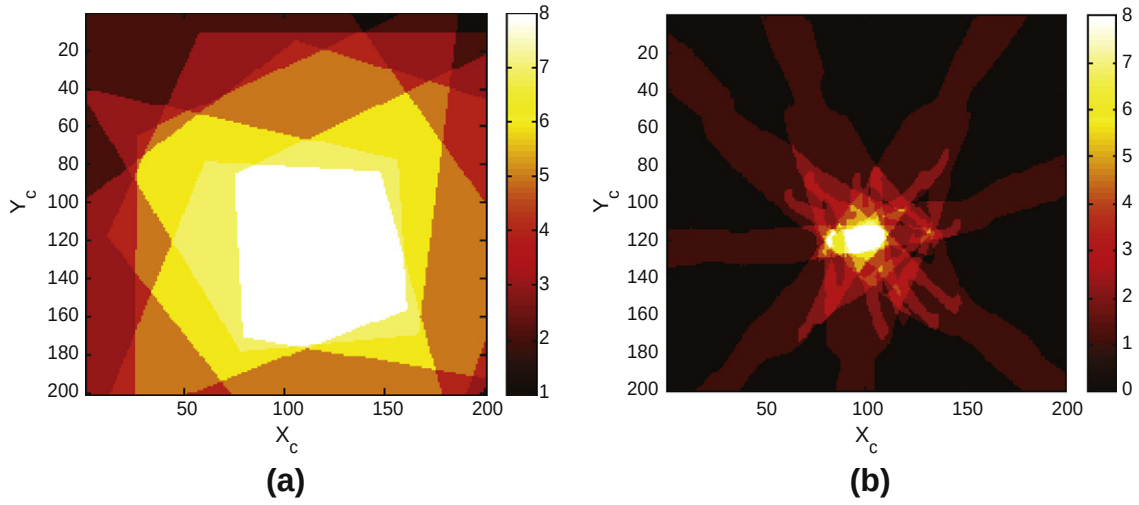


Fig. 2. (a) A slice of the grid G for $Z_c = 0$ cm. Different gray level values denote scene regions of variable visibility from a multiview configuration of eight cameras. Dark regions are visible from one view while bright regions are visible from all the cameras. (b) A slice of the grid G is shown (for $Z_c = 100$ cm). Each view projects on this slice its captured silhouette. White areas correspond to silhouette intersections from all the views of the configuration.

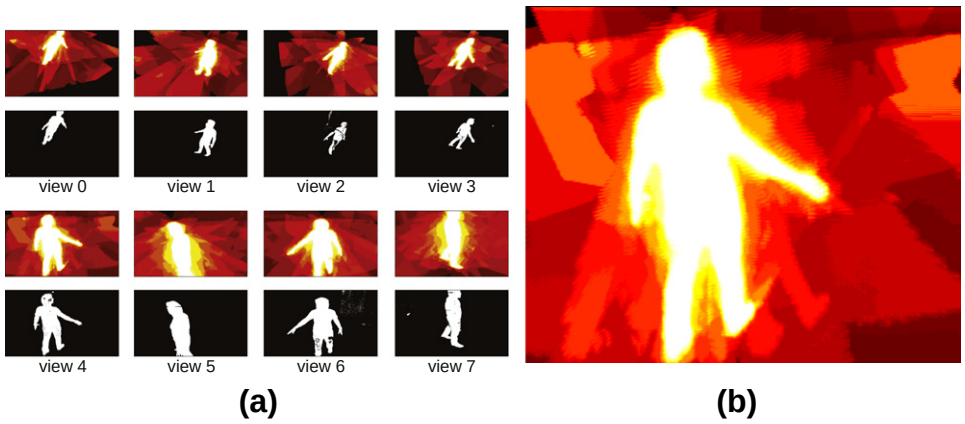


Fig. 3. (a) The confidence maps and silhouette images for a single frame across all views of an 8-camera configuration. In (b), the confidence map obtained in view 4 is shown in greater detail. Brighter colors correspond to higher confidence values.

vidual camera foreground detections, it is very unlikely that a strong consensus is built around them. Thus, confidence maps represent more robustly the segmentation of a scene into foreground and background, compared to the single view silhouette estimates.

4. Particle Swarm Optimization

Classical approaches on solving optimization problems are often based on the evaluation of the derivatives of the defined objective function. In real-world optimization problems, the analytical expression of the objective function is not known or it is multimodal, i.e., has several local minima. Additionally, its derivatives may not-even be defined at certain points of the parameter space. To cope with such problems, derivative-free optimization algorithms have been proposed. One such approach is Particle Swarm Optimization (PSO) [14]. PSO is a population based stochastic optimization method that utilizes swarm intelligence to find extrema of nonlinear continuous functions (a.k.a. *objective* or *fitness* functions). It is similar to other evolutionary techniques like *Genetic Algorithms* [13] with the major difference of having no crossover and mutation operators. PSO exhibits better performance compared to several other optimization methods [1] and is very efficient in terms of computational cost.

Particle Swarm Optimization is an attractive optimization method for the problem at hand for several reasons. It performs well with non-smooth, multimodal objective functions and requires a relatively low number of objective function evaluations [1]. It depends on a very few parameters and it scales well with the number of parameters to be optimized. Finally, it is inherently parallel, leaving room for parallel implementations that can drastically reduce the computation time required for optimization, especially when this is intended to be performed on-line.

4.1. Social optimization

PSO is based on social interactions between the atoms of a population in order to optimize a problem modeled with a specific fitness function. The method is inspired by the social behavior exhibited in flocks of birds and schools of fishes. As such, it handles populations of particles that are defined in the optimization space. A social network between individuals (i.e., particles) is defined. The particles are candidate solutions that are initialized randomly. The social network determines the interactions that can take place (e.g., particles can only interact with their neighbors). During the execution of the PSO algorithm, particles evaluate the fitness of the candidate solutions that represent and store in memory the

parameters achieving the optimum fitness values. Moreover, they adjust their velocities through predefined update equations. Finally, they move in the parameter space, i.e., update their positions according to a random linear blending performed upon two velocity vectors. One of these vectors points towards the particle's local best solution and the other towards the best solution in a neighborhood of particles. This process evolves iteratively, where each iteration is called a *generation*, until a termination criterion is met. Such criteria include the convergence of the whole or of a portion of the particle population to a single solution, the execution of an upper bound of iterations, the achievement of a specific fitness score, etc.

A great number of PSO variants have been proposed. In this work, the simplest form of the PSO algorithm, called *canonical* PSO [7] has been employed. Other popular variants include the *fully informed* PSO [19] as well as variants that define dynamic neighborhood topologies [25] and those that utilize enhanced diversity at updating [2]. Variants have also been defined by using heuristic velocity update rules or by explicitly handling discrete optimization problems [6].

4.2. Canonical PSO

In canonical PSO, the topology of the population reduces to only one neighborhood. Following the notation introduced in [27], every particle holds its current position (current candidate solution, set of parameters) in a vector x_t and its current velocity in a vector v_t. Moreover, each particle stores in vector p_i the position at which it achieved, up to the current generation t, the highest fitness score. Finally, the swarm as a whole, stores in vector p_g the best position encountered across all particles of the swarm. p_g is broadcasted to the entire swarm, so every particle is aware of the current global optimum. The update equations that are applied in every generation t to reestimate the particle velocities and positions are

$$v_t = K(v_{t-1} + c_1 r_1(p_i - x_{t-1}) + c_2 r_2(p_g - x_{t-1})) \tag{15}$$

and

$$x_t = x_{t-1} + v_t, \tag{16}$$

where K is a constant *constriction factor* [5] defined as

$$K = \frac{2}{\left|2 - \psi - \sqrt{\psi^2 - 4\psi}\right|}, \quad \psi = c_1 + c_2. \tag{17}$$

In Eqs. (15) and (17), c_1 is called the *cognitive component*, c_2 is termed the *social component* and r_1,r_2 are random samples of a uniform distribution in the range [0,1]. Finally, $c_1 + c_2 > 4$ must hold [5]. In all performed experiments the values $c_1 = 2.8$ and $c_2 = 1.3$ were used.

As mentioned earlier, the particles are initialized at random positions and their velocities are initialized to zero. Each dimension of the multidimensional parameter space is bounded in some range. If, during the position update, a velocity component forces the particle to move to a point outside the bounded search space, this component is zeroed and the particle doesn't perform any move at the corresponding dimension.

5. Optimization of foreground detection parameters

The proposed algorithm is an iterative procedure that utilizes canonical PSO to search for the optimal parameter vector across the parameter space of the foreground detection algorithm presented in Section 2. The optimal parameter vector is defined to be the one that maximizes the similarity between silhouettes and confidence maps across all available views. Each particle position corresponds to a set of foreground detection parameter values. During particle evaluation, a foreground detection instance is initialized using the particle's position and applied to an image subsequence to produce a set of silhouette estimates. The use of sequences instead of single frames is mandatory because, by definition, the foreground detection algorithm requires a history of observations in order to produce reliable results.

The proposed iterative optimization process consists of the following steps (a) calculation of the confidence maps based on the current silhouette estimates, (b) optimization of the foreground segmentation parameters using the computed confidence maps, and (c) calculation of new silhouette estimates using the optimized parameters. By iterating the above steps in a closed loop, both the estimated parameters and the quality of the produced silhouettes get improved. A similar idea in the field of Machine Learning is employed in the principle of *generalized policy iteration* [26]. The defined fitness function measures the similarity between confidence maps and silhouette estimates across a given image sequence and for every view. Silhouette estimates are computed as reported in Section 2 from an instance of the foreground detection algorithm that is initialized by the position vector of a given particle. Confidence maps are produced by the silhouette estimates and the additional calibration information of the multiview configuration, as detailed in Section 3.2.

More specifically, let $S_i^t(x,y)$ denote a point of the silhouette image of frame t captured by camera i. Let also $\mathscr{C}_i^t(x,y)$ denote the value of the confidence map for the same point. The distance $D_{A,i,t}$ between silhouettes and confidence maps for a set of points A is calculated as:

$$D_{A,i,t} = \sum_{(x_p,y_p)\in A} \left|S_i^t(x_p,y_p) - \mathscr{C}_i^t(x_p,y_p)\right|. \tag{18}$$

If we denote with $P_{fg,i}^t$ the set of silhouette pixels of frame t of view i (i.e., foreground pixels) and with $P_{bg,i}^t$ the set of the background pixels, then the fitness function is defined as

$$F = \sum_{t\in[0,T]} e^{(1-r_t/2|C|)}, \tag{19}$$

where

$$r_t = \sum_{j\in C}\left(\frac{D_{P_{fg,j}^t,j,t}}{\left|P_{fg,j}^t\right|} + \frac{D_{P_{bg,j}^t,j,t}}{\left|P_{bg,j}^t\right|}\right). \tag{20}$$

Algorithm 1 provides a summary of the computation of the fitness function while Algorithm 2 provides a summary of the full optimization process.

Algorithm 1. Computation of the fitness function

Input: Particle $\mathscr{P}, T, N_c$
Output: Fitness score F
$F = 0$
foreach $l = 1,2,\ldots,T$ **do**
  Compute silhouettes $S_i, \forall i \in [1,N_c]$ (as described in Sec. 2);
  Compute confidence maps $\mathscr{C}_i, \forall i \in [1,N_c]$ (Eq. (11));
  Compute $P_{fg,i,l}$, $P_{bg,i,l}$, $\forall i \in [1,N_c]$;
  $r_l = \sum_{i\in[1,N_c]}\left(\frac{D_{P_{fg,i,l}}}{|P_{fg,i,l}|} + \frac{D_{P_{bg,i,l}}}{|P_{bg,i,l}|}\right)$ (Eq. (20));
  $F = F + e^{(1-r_l/2|C|)}$;
return (F);

6. Experiments

The goal of the performed experiments is (a) to show whether the proposed method can be applied successfully to image sequences acquired by a calibrated multiview configuration in order

to automatically tune the foreground detection parameters and produce optimal silhouette images in a totally unsupervised manner and (b) to investigate the influence of several factors (i.e., PSO parameters, noise level, camera number and topology, etc.) on the quality of the obtained results.

6.1. Parameter selection

The performance of foreground detection is governed by the learning rate parameter α (Eqs. (2)–(4)) that determines the speed of the adaptation. A uniform update speed is enforced by setting $\alpha = 1/T$.

Algorithm 2. Optimization of the foreground detection parameters.

Input: Number of PSO generations N_g, length of frame sequence T, PSO population size N_p, number of cameras N_c
Outpur: Optimal foreground detection parameter vector $\overline{\mathscr{P}^*}$
$F_{max} = 0$;
Initialize $N_c \times N_p$ particles p_i randomly (random x_i, $v_i = 0$);
foreach $n = 1,2,\ldots,N_g$ **do**
  Perform particle p_i flight, $\forall i \in [1, N_c \times N_p]$ (Eq. (15));
  Compute fitness F_i of p_i,$\forall i \in [1, N_c \times N_p]$ (through Algorithm 1);
  if $F_i > F_{max}$ **then**
    $F_{max} = F_i$;
    $\overline{\mathscr{P}^*} = p_i$;
  Update velocity of p_i, $\forall i \in [1, N_c \times N_p]$ (Eq. (16));
return ($\overline{\mathscr{P}^*}$)

Another important parameter is the threshold T_b on the squared Mahalanobis distance upon which it is decided if a given sample is close to a background GMM component or not. It must be noted that T_b is different from the threshold T_g that specifies whether a sample belongs to any of the mixture components modeling either background or foreground. According to Zivkovic [29], typical values are $T_b = 16\sigma_m^{(t)}$ and $T_g = 3\sigma_m^{(t)}$ for the mth component at time step t.

In general, it is proposed that a total of four Gaussian components are sufficient for the purposes of foreground detection. Therefore, in our experiments this parameter did not vary. Moreover, let $T_b = 1 - c_f$ where it holds that $0 \leqslant T_b \leqslant 1$. The threshold T_b determines (see Eqs. (5) and (6)) the number of mixture components that model the background. In order for the background modeling to be valid, T_b must have a value that allows for the background to be modeled by at least one Gaussian component. Typically, $c_f = 0.1$ which leads to $T_b = 0.9$. Finally, the initial variance σ_0 of the newly imported components in the mixture, influences the speed of adaptation. A typical value for this parameter is $\sigma_0 = 10$.

As the parameters $\{\alpha, T_b, T_g, \sigma_0\}$ have a great impact on the final result of the foreground detection process, they were selected as the target variables of the proposed optimization process.

6.2. Experimental setup

The experimental validation of the proposed method was based on two datasets. The first is the "Dancer" dataset[1] of Inria's 4D repository. This dataset captures the movements of a female dancer through a configuration of eight calibrated cameras. Each view captures 251 synchronized frames of size 780×582. The first 50 frames contain only scene background and are provided for proper initialization of the background modeling process. From those, 49 frames were omitted, so the resulting sequence starts with a single frame showing the scene background in isolation. On top of the actual data, the dataset comes with a set of preprocessed silhouettes (one per frame). These data are not used in the optimization process but form a basis for the quantitative evaluation of the non-supervised foreground detection algorithm.

The second dataset[2] is a synthetic, noise-free dataset, showing a 3D rendered model of a Kung-Fu girl in action. This has been acquired by a virtual multiview setup of 25 cameras and contains 201 synchronized frames of image size 320×240. There is a single frame showing the scene background in isolation. This dataset also comes with a ground truth set of silhouettes that is produced automatically by rendering the 3D model with no lights, resulting in a white silhouette on a black background.

The description of the foreground detection method in [29] suggests a parameter set that performs relatively well in the general case. We refer to these parameters as *typical parameters*. Throughout our experiments, we evaluate the typical parameters and the parameters suggested by our methodology against the available ground truth. This evaluation involves a comparison of the silhouette images produced by a set of parameters against the available ground truth silhouettes. More specifically, let Ω be the set of all image pixels for all cameras and time instances. Then the measure used for comparing the resulting silhouette images to ground truth is:

$$q = 1 - \frac{D_\Omega}{|\Omega|}, \tag{21}$$

where

$$D_\Omega = \sum_{t\in[0,T]} \sum_{(x,y)\in\Omega} |S^t(x,y) - T^t(x,y)| \tag{22}$$

and $T^t(x,y)$ denotes the ground truth available for point (x,y) at time t. A value of $q = 1$ signifies silhouette images identical to the ground truth and, therefore, perfect foreground detection parameters.

6.3. Dancer dataset

We present quantitative and qualitative results obtained from the application of the proposed method on the dancer dataset. As detailed in Section 6.1, the most critical foreground detection parameters are α, T_b, T_g and σ_0. In a first experiment, we used a population of 15 particles running PSO for 50 generations on the 4D parameter space $\{\alpha, T_b, T_g, \sigma_0\}$. Each particle is evaluated on the entire dancer sequence. We call this the *exhaustive* or the *all-frames experiment*.

In a second experiment, the self-evaluation of each particle considered only the first 10 frames of the entire sequence. We call this experiment the *10-frames experiment*. In this case, a population of eight particles run PSO for 20 generations on the 3D parameter space of $\{T_b, T_g, \sigma_0\}$. The reason for excluding parameter α is that a value of α that is optimal on the small, 10-frames time window, cannot generalize well in a sequence of extended length. Therefore, α was fixed to the typical value while PSO was set to jointly optimize parameters T_b, T_g and σ_0.

The parameter vectors estimated in the two experiments were evaluated on the entire sequence. The typical parameter vector as well as the initialization parameter vector of the second experiment were also evaluated. These four parameter vectors are listed in Table 1. Table 1 also reports the mean fitness values across the whole frame range of the sequence achieved by each parameter

[1] Available for download at http://charibdis.inrialpes.fr/public/viewgroup/1.

[2] Available for download at http://www.mpi-inf.mpg.de/departments/irg3/kungfu/.

Table 1
Evaluation of foreground detection parameters on the dancer sequence.

Parameter set	T_b	T_g	σ_0	α	Mean fitness value
Typical	16.0	9.0	11.0	0.0001	2283.29
Initialization	3.3	13.2	2.7	0.0001	211.71
10-frames best	18.6	19.0	37.0	0.0001	6389.85
All-frames best	11.9	16.7	41.6	0.0004	**6613.91**

set. The detailed fitness graph for all parameter sets in all sequence frames is shown in Fig. 4.

From these results, it can be verified that the parameter set returned from the exhaustive experiment was the best, followed by the parameter set resulting from the 10-frames experiment. Those two sets achieved far better fitness scores than the typical parameters, having a marginal difference between them. Fig. 5 shows how those fitness scores translate to ground truth similarity.

Two important conclusions can be drawn from these results. First, there is a consistency between fitness function scores and ground truth similarity scores, thus the fitness function is well defined. Second, the results of the exhaustive experiment are very similar to the results of the 10-frames experiment, leading to the conclusion that the parameters found on the small training set generalize very well for the rest of the sequence assuming that there are no major changes in the environment. This is an important observation that can be exploited to avoid the significant additional computational overhead of optimizing a large population of particles across many generations on the whole sequence at a small quality pay-off. This also demonstrates that the proposed method can be used for the automatic tuning of parameters on streaming sequences using just a small number of frames to estimate the proper parameters. Examples of the silhouette images produced by applying the four different instances of foreground detection on a specific view and frame together with the ground truth are shown in Fig. 6. As it can be verified, the noise patterns appearing in the images corresponding to the initialization and typical parameter sets are missing from the image corresponding to the optimal parameter set.

We furthermore isolated the particle that returned the optimal position for the 10-frames experiment and we recorded its route to this solution. The fitness of this particle as a function of generations is illustrated in Fig. 7. The plot indicates that the proposed method requires approximately 15 generations to optimize the parameters.

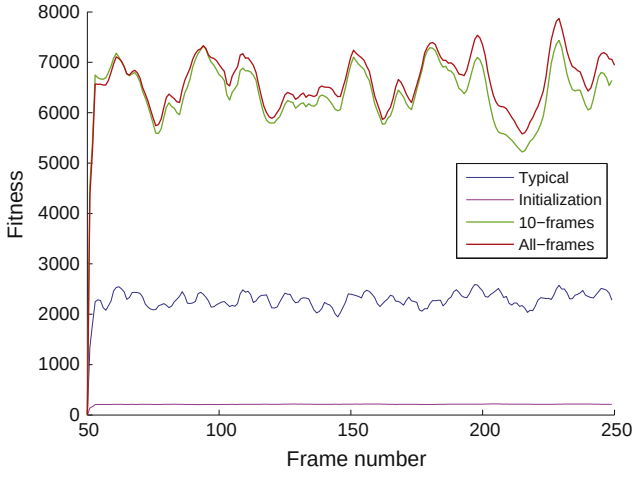


Fig. 4. Fitness curves of the four parameter vectors for the dancer dataset experiments.

6.4. Kung-Fu girl dataset

We also conducted the 10-frames experiment on the Kung-Fu girl dataset (8 particles, 20 generations, training set of 10 frames, $\{T_b, T_g, \sigma_0\}$ parameter space). Following the same approach as in the case of the dancer dataset, we evaluated the three parameter vectors shown in Table 2. The corresponding fitness graphs are shown in Fig. 8a. Similarity to ground truth was computed as shown in Fig. 8b. Finally, examples of silhouette images from the three detection instances that correspond to the parameter sets of the experiments are shown in Fig. 9.

6.5. Noise effects

The presence of noise in the input image is responsible for increasing the number of detected foreground pixels. This is because color variations due to noise are more likely to manifest themselves as foreground rather than as a significantly varying background. This can be observed on the output of the typical parameters for the dancer dataset (Fig. 6) where foreground pixels are distributed, following a certain camera dependent noise pattern, across the entire image area. Thus, in the dancer sequence experiments, the proposed optimization seeks the optimal parameter set that also compensates for image noise. In the case of the experiment with the synthetic, noise-free Kung-Fu girl dataset, the algorithm just optimizes the similarity to the ground truth.

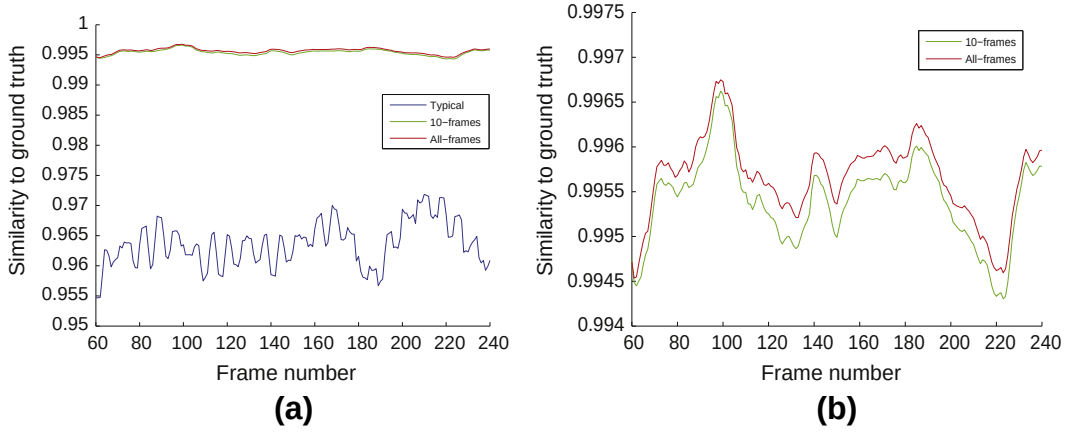


Fig. 5. (a) Comparison of the silhouettes produced by each parameter vector to the ground truth, (b) the performance of the 10-frames and exhaustive experiments, isolated.

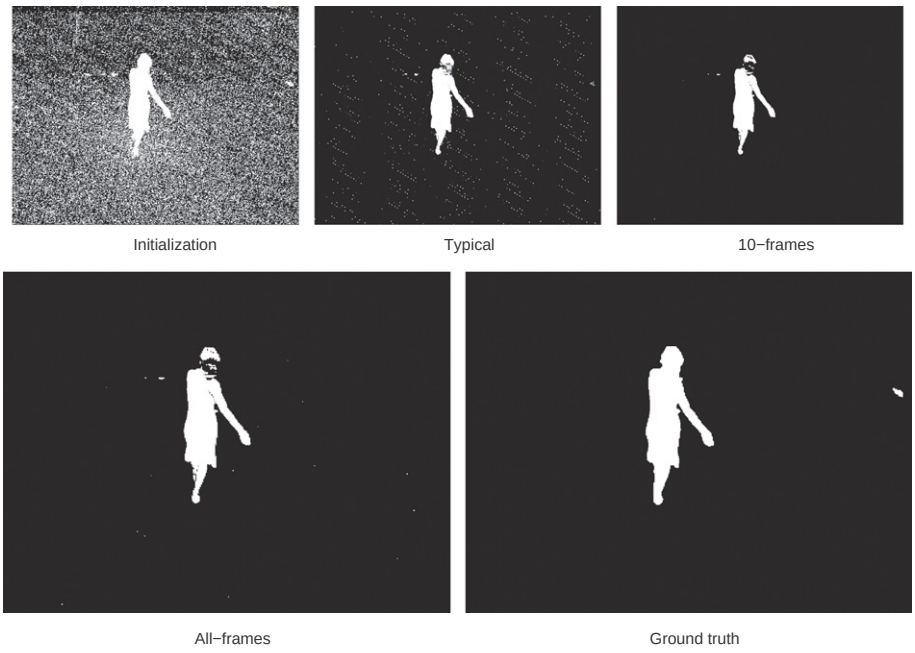


Fig. 6. Example silhouettes calculated by the foreground detection algorithm for frame #110 of the dancer sequence as shown from camera #3 and for the four different parameter sets. The ground truth silhouette is also provided as a reference.

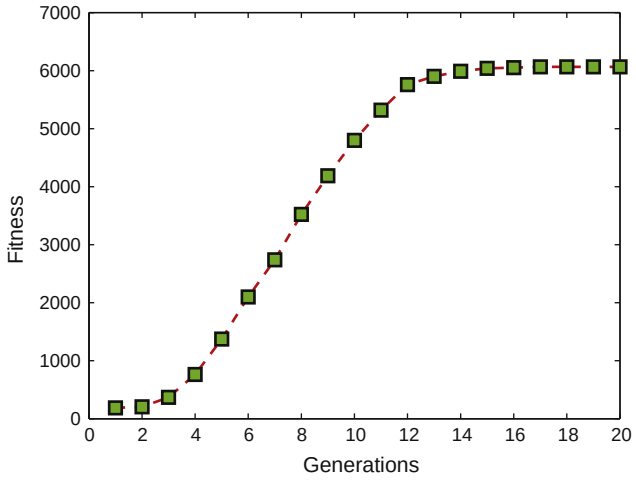


Fig. 7. The route of the initialization parameter set to the optimal position. Markers are placed at the fitness scores that the particle achieved at each generation.

Table 2
Evaluation of foreground detection parameters on the Kung-Fu girl dataset.

Parameter	T_b	T_g	σ_0	Mean fitness value
Typical	16.00	9.00	11.00	4322.30
Initialization	39.50	14.20	27.40	805.99
10-frames best	3.05	2.60	1.30	**10489.90**

As it is shown in Fig. 9, the silhouette image produced with the best parameter set is almost identical to the ground truth, without any holes. On the contrary, the corresponding result for the dancer sequence contains some holes, as a result of the presence of noise.

A series of experiments were conducted to systematically measure the behavior of the proposed algorithm to various noise levels. In these experiments we contaminated the original Kung-Fu girl dataset with three different levels of Gaussian noise ($\mu_1 = 0, \sigma_1^2 = 0.0001$), ($\mu_2 = 0, \sigma_2^2 = 0.00025$), ($\mu_3 = 0, \sigma_3^2 = 0.0005$). Next, we conducted the 10-frames experiment on the resulting datasets and evaluated the results. Finally, we compared the results against the typical parameters. Mean fitness values for the various noise levels are shown in Table 3. Fitness and similarity graphs are shown in Fig. 10, while silhouette examples with the corresponding ground truth are shown in Fig. 11.

As it can be verified, although parameter optimization is affected by noise, in all three cases the suggested parameters result in silhouettes closer to the ground truth than those produced by the typical parameter set. Moreover, for the case where $\sigma_1^2 = 0.0001$, we found that the typical parameters were very close to the optimal parameters returned by the optimization procedure (see Table 4). This might serve as an indication that the typical parameters are tuned to deal with this particular level of image noise. Another interesting observation is that as the noise level increases, the optimization method automatically, but also reasonably, increases the parameter σ_0.

6.6. Camera placement and number of cameras

Another interesting problem dimension is the variability of the obtained results with respect to the placement of the available cameras and their number. The topology of the camera network highly influences the results. More specifically, the method fails to optimize the foreground detection parameters if the cameras arrangement does not permit the accurate voting in the voxel space. As an example, consider a configuration where cameras are placed in one side of the foreground object, only. The fact that large parts of the foreground object are not visible by any of the cameras results in a voxel space that does not accurately represent the object's 3D structure. This produces inaccurate confidence maps which, in turn, leads the parameter optimization process far from its optimal values.

Provided that the cameras are placed in a way that surrounds the foreground objects, the increase of the number of cameras does not improve considerably the obtained results. In order to examine the effects of the number of cameras on the performance of the

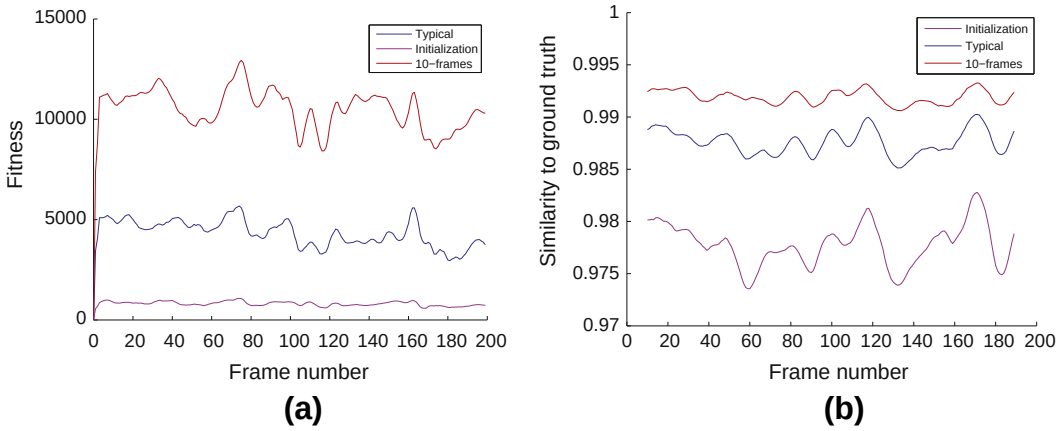


Fig. 8. (a) Fitness curves for the Kung-Fu girl experiments, (b) comparison of the silhouettes produced by each parameter vector to the ground truth for the Kung-Fu girl dataset.

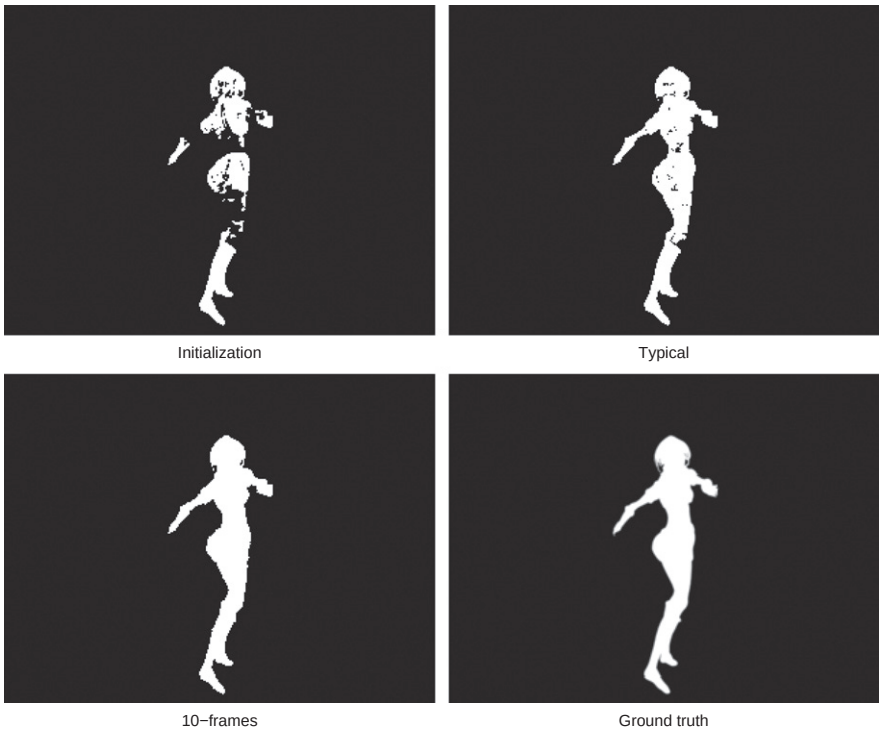


Fig. 9. Silhouettes calculated by the foreground detection algorithm for frame #127 of the Kung-Fu sequence (camera #14) for the three parameter vectors. The ground truth silhouette is also provided.

Table 3
Mean fitness values for various levels of noise contamination of the Kung-Fu girl sequence.

Parameter set	σ_1^2	σ_2^2	σ_3^2
Typical	3661.52	412.25	231.62
10-frames best	**3725.85**	**1473.76**	**859.30**

method, we conducted experiments on the Kung-Fu girl dataset, each time utilizing a different camera subset of the original 25-camera configuration. Sixteen out of the 25 views have nodal points arranged on a circle and optical axes pointing towards the center of this circle. We considered 11 different camera subsets with a number of cameras ranging between 6 and 16. In each case, cameras were distributed as evenly as possible over the entire circle. For all the 11 configurations tested, the resulting fitness value remained practically unchanged and equal to the one reported in the full 25 cameras experiment presented in Section 6.4. Analogous experiments with the dancer data set led to exactly the same performance.

6.7. Optimization of individual camera parameters

In previous experiments a single parameter vector is optimized and used for the entire camera set. This vector defines a low dimensional search space for the optimization algorithm. It is known that the canonical PSO algorithm performs very efficiently in such low dimensional spaces where it only needs to utilize a

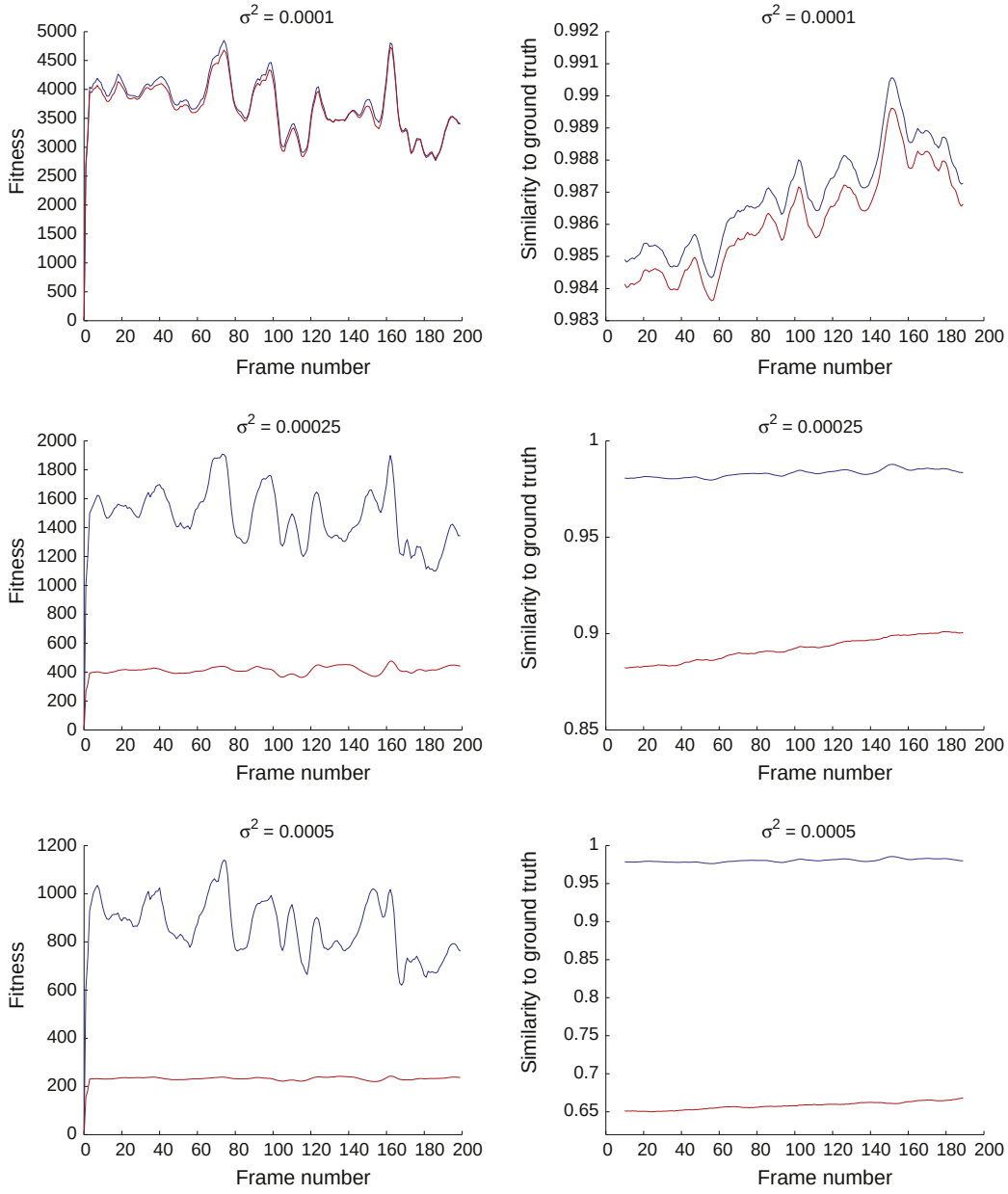


Fig. 10. Fitness graphs for the 10-frames and the typical parameters experiments on the Kung-Fu girl dataset for various noise levels (left) and comparison with the ground truth (right). Blue curves correspond to the 10-frames experiments and red curves to the typical parameters set.

small population of particles for few generations. We have further examined the behavior of the proposed method on larger search spaces. In a series of experiments the optimization method was employed to optimize individual camera parameters. More specifically, for an n camera setup, the parameter vectors had a dimension of $3n$.

On the dancer dataset the total number of parameters to optimize formed a vector of 24 dimensions (i.e., 8 cameras, 3 parameters per camera). The optimization procedure for this experiment, utilized 8 particles for 20 generations. The fitness and similarity-to-ground-truth curves found to be identical to the ones produced by the 10-frames experiment that was described in Section 6.3. A similar experiment was also conducted for the Kung-Fu girl dataset where 16 cameras were utilized resulting in a total parameter vector of 48 dimensions. For this experiment, the optimization algorithm required 200 generations of an 8-particle population to converge to results similar to the ones presented in Section 6.4.

6.8. Implementation and computational performance issues

The experiments were conducted on a PC with 6GB RAM, Intel 920 core i7 CPU and a Nvidia GTX 295 GPU. Confidence map computation and multiview silhouette estimates were implemented on GPU, using Nvidia's CUDA framework.[3] For the dancer dataset confidence maps were calculated at a rate of roughly 250 frames per second while on the Kung-Fu girl dataset we reached a rate of 800 frames per second. For foreground detection we employed the publicly available[4] CPU implementation of the method described in [29]. Foreground detection calculations for one generation of 8 particles and for the 10-frames experiment on the dancer dataset took 62 s. On the Kung-Fu girl dataset the corresponding time was 23 s.

[3] http://developer.nvidia.com/object/cuda_2_3_downloads.html.
[4] http://staff.science.uva.nl/zivkovic/Publications.

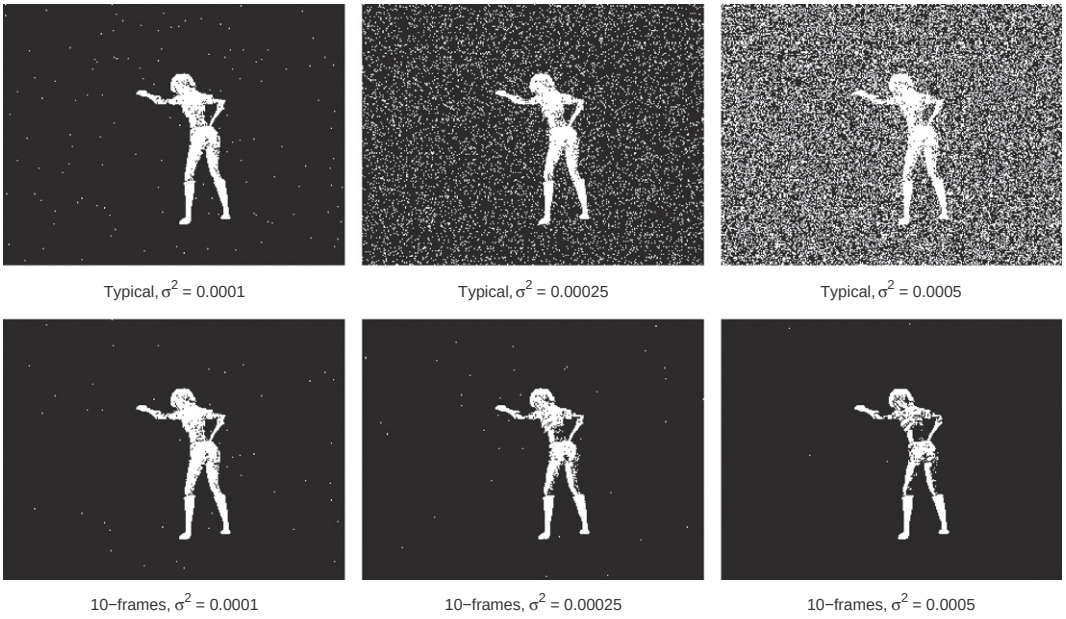


Fig. 11. The silhouettes returned by the typical and and 10-frames best foreground detection instances for the three noise levels of the conducted experiments. Results correspond to frame #138, view 8 of the Kung-Fu girl dataset.

Table 4
Parameter vectors estimated for the Kung-Fu dataset at various noise levels.

Parameter set	T_b	T_g	σ_0
Typical	16	9	11
$\sigma_1^2 = 0.00010$	16.8	19.5	11.0
$\sigma_2^2 = 0.00025$	33.5	23.7	13.0
$\sigma_3^2 = 0.00050$	34.8	30.2	28.0

7. Conclusions

We presented a novel algorithm for optimizing, in an unsupervised manner, the foreground detection parameters of a calibrated multicamera configuration. The proposed method successfully exploits information regarding the consensus of the setup on what constitutes foreground in an observed scene. By encoding this information in a fitness function, Particle Swarm Optimization optimizes the foreground detection parameters. Results showed a strong correlation of the fitness curve with the similarity-to-ground-truth curve, leading to the conclusion that the proposed definition of the fitness function is a good choice for the specific task. It was also shown that this method can be used, provided an efficient foreground detection implementation, for online applications.

The most important advantage of the proposed algorithm is that it does not require prior ground truth information or other kind of supervision. As such, it can be used as a tool for automatically adjusting the foreground detection parameters in frequently changing environments. The data used to evaluate our method have been captured in laboratory conditions. This has been motivated by the availability of ground-truth for these data sets, which is required for the quantitative evaluation of the proposed approach. It is expected that the benefits from the application of the proposed method in uncontrolled environments (i.e. outdoors surveillance) will be much greater due to the fact that, in such conditions, there is no single parameter set that performs well on average. Current and future work includes the extension of this approach to other interesting problems where multiview consensus can be exploited towards relaxing the requirement for ground truth data and/or supervision.

References

[1] P.J. Angeline, Evolutionary optimization versus particle swarm optimization: philosophy and performance differences, Evolutionary Programming VII, LNCS 1447 (1998) 601–610.
[2] T.M. Blackwell, P.J. Bentley, Don't push me collision-avoiding swarms, in: IEEE Conference on Evolutionary Computation, 2002, pp. 1691–1696.
[3] T. Bouwmans, F. El Baf, B. Vachon, Background modeling using mixture of gaussians for foreground detection – a survey, Recent Patents on Computer Science 1 (3) (2008) 219–237.
[4] K.M. Cheung, S. Baker, T. Kanade, Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture, in: IEEE Conference on Computer Vision and Pattern Recognition, 2003.
[5] M. Clerc, The swarm and the queen: towards a deterministic and adaptive particle swarm optimization, in: Congress on Evolutionary Computation, vol. 3, 1999, pp. 1951–1957.
[6] M. Clerc, Discrete Particle Swarm Optimization. New Optimization Techniques in Engineering, Springer, Verlag, 2004.
[7] M. Clerc, J. Kennedy, The particle swarm-explosion, stability and convergence in a multidimensional complex space, IEEE Transactions on Evolutionary Computation 6 (1) (2002) 58–73.
[8] R.T. Collins, A.J. Lipton, H. Fujiyoshi, T. Kanade, Algorithms for cooperative multisensor surveillance, in: Proceedings of the IEEE, 2001.
[9] W. Du, J.B. Hayet, J. Piater, J. Verly, Collaborative multi-camera tracking of athletes in team sports, in: Computer Vision Based Analysis in Sport Environments (CVBASE), 2006, pp. 2–13.
[10] A. Elgammal, D. Harwod, L.S. Davis, Non-parametric model for background subtraction, in: International Conference on Computer Vision, FRAME-RATE Workshop, 1999.
[11] J.S. Franco, E. Boyer, Fusion of multiview silhouette cues using a space occupancy grid, in: International Conference on Computer Vision, vol. 2, 2005, pp. 1747–1753.
[12] B. Han, D. Comaniciu, L. Davis, Sequential kernel density approximation through mode propagation: applications to background modeling, in: Asian Conference on Computer Vision, 2004.
[13] J.H. Holland, Adaptation in Natural and Artificial Systems, The University of Michigan Press, 1975.
[14] J. Kennedy, R. Eberhart, Particle swarm optimization, in: IEEE International Conference on Neural Networks, 1995, pp. 1942–1948.
[15] A. Laurentini, The visual hull concept for silhouette-based image understanding, IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (2) (1994) 150–162.
[16] W.N. Martin, J.K. Aggrawal, Volumetric descriptions of objects from multiple views, IEEE Transactions on Pattern Analysis and Machine Intelligence 5 (2) (1983) 150–158.
[17] T. Matsuyama, N. Ukita, Real-time multi-target tracking by a cooperative distributed vision system, in: Proceedings of the IEEE, 2002, pp. 1136–1150.
[18] T. Matsuyama, X. Wu, T. Takai, T. Wada, Real-time 3d shape reconstruction, dynamic 3d mesh deformation, and high fidelity visualization for 3d video, Computer Vision and Image Understanding 96 (3) (2004) 393–434.

[19] R. Mendes, J. Kennedy, J. Neves, The fully informed particle swarm: simpler, maybe better, IEEE Transactions on Evolutionary Computation 8 (3) (2004) 204–210.
[20] N.M. Oliver, B. Rosario, A.P. Pentland, A bayesian computer vision system for modeling human interactions, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8) (2000) 831–843.
[21] M. Piccardi, Background subtraction techniques: a review, in: IEEE Conference on System, Man and Cybernetics, vol. 4, 2004, pp. 3099–3104.
[22] C.J. Van Rijsbergen, Information Retrieval, Butterworth Heinemann, Newton, MA, USA, 1979.
[23] A. Smolic, K. Mueller, P. Merkle, C. Fehn, P. Kauff, P. Eisert, T. Wiegand, 3d video and free viewpoint video-technologies, applications and mpeg standards, in: IEEE Conference on Multimedia and Expo, 2006, pp. 2161–2164.
[24] C. Stauffer, W. Grimson, Adaptive background mixture models for real-time tracking, in: IEEE Conference of Computer Vision and Pattern Recognition, vol. 2, 1999, pp. 246–252.
[25] P.N. Suganthan, Particle swarm optimiser with neighborhood operator, in: IEEE Conference on Evolutionary Computation, 1999, pp. 1958–1962.
[26] R. S Sutton, A.G. Barto, Reinforcement Learning: An Introduction, MIT Press, 1998.
[27] B. White, M. Shaw, Automatically tuning background subtraction parameters using particle swarm optimization, in: IEEE Conference on Multimedia and Expo, 2007, pp. 1826–1829.
[28] C. Wu, H. Aghajan, Collaborative gesture analysis in multi-camera networks, in: ACM SenSys Workshop on DSC, 2006.
[29] Z. Zivkovic, Improved adaptive gaussian mixture model for background subtraction, in: International Conference on Pattern Recognition, 2004.

From multiple views to textured 3D meshes: a GPU-powered approach

K. Tzevanidis, X. Zabulis, T. Sarmis, P. Koutlemanis,
N. Kyriazis, and A. Argyros

Institute of Computer Science (ICS)
Foundation for Research and Technology - Hellas (FORTH)
N.Plastira 100, Vassilika Vouton, GR 700 13
Heraklion, Crete, GREECE
{ktzevani, zabulis, argyros}@ics.forth.gr

Abstract. We present work on exploiting modern graphics hardware towards the real-time production of a textured 3D mesh representation of a scene observed by a multicamera system. The employed computational infrastructure consists of a network of four PC workstations each of which is connected to a pair of cameras. One of the PCs is equipped with a GPU that is used for parallel computations. The result of the processing is a list of texture mapped triangles representing the reconstructed surfaces. In contrast to previous works, the entire processing pipeline (foreground segmentation, 3D reconstruction, 3D mesh computation, 3D mesh smoothing and texture mapping) has been implemented on the GPU. Experimental results demonstrate that an accurate, high resolution, texture-mapped 3D reconstruction of a scene observed by eight cameras is achievable in real time.

1 Introduction

The goal of this work is the design and the implementation of a multicamera system that captures 4D videos of human grasping and manipulation activities performed on a desktop environment. Thus, the intended output of the target system is a temporal sequence of texture mapped, accurate 3D mesh representations of the observed scene. This constitutes rich perceptual input that may feed higher level modules responsible for scene understanding and human activity interpretation.

From the advent of GPU programmable pipeline, researchers have made great efforts to exploit the computational power provided by the graphics hardware (i.e. GPGPUs). The evolution of GPUs led to the introduction of flexible computing models such as shader model 4.0 and CUDA that support general purpose computations. Various GPU implementations of shape-from-silhouette reconstruction have been presented in the recent literature [1, 2]. Moreover, following past attempts on real-time reconstruction and rendering (e.g. [3, 4]), some recent works introduce full 3D reconstruction systems [5, 6] that incorporate modern graphics hardware for their calculations. The later implementations take as input

segmented object silhouettes and produce as output voxel scene representations. In contrast to these systems, the one proposed in this paper parallelizes the whole processing pipeline that consists of foreground object segmentation, visual hull computation and smoothing, 3D mesh calculation and texture mapping. The algorithms implementing this processing chain are inherently parallel. We capitalize on the enormous computational power of modern GPU hardware through NVIDIA's CUDA framework, in order to exploit this fact and to achieve realtime performance.

The remainder of this paper is organized as follows. Section 2 introduces the system architecture both at hardware and software level. Section 3 details the GPU-based parallel implementation of the 3D reconstruction process. Experiments and performance measurements are presented in Sec. 4. Finally, Sec. 5 provides conclusions and suggestions for future enhancements of the proposed system.

2 Infrastructure

2.1 Hardware Configuration

The developed multicamera system is installed around a $2 \times 1m^2$ bench and consists of 8 *Flea2* PointGrey cameras. Each camera has a maximum framerate of 30 fps at highest (i.e. 1280×960) image resolution. The system employs four computers with quad-core Intel i7 920 CPUs and 6 GBs RAM each, connected by an 1 Gbit ethernet link. Figure 1 shows the overall architecture along with a picture of the developed multicamera system infrastructure.

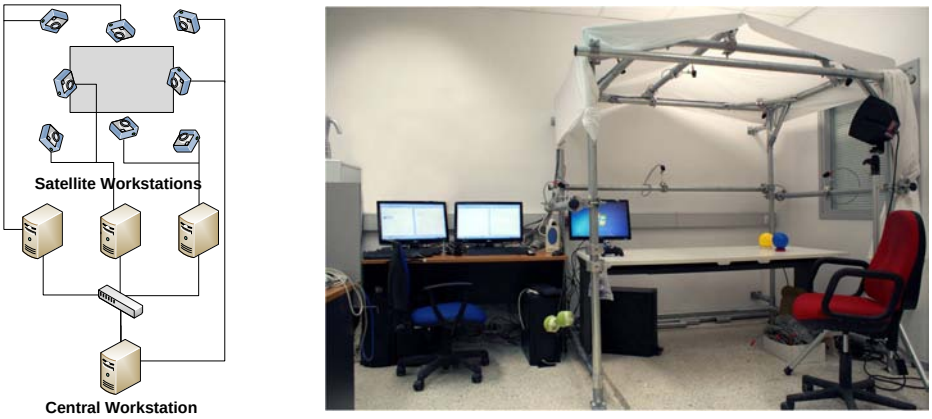
In our *switched-star* network topology, one of the four computers is declared as the *central workstation* and the remaining three as the *satellite workstations*. The central workstation's configuration, includes also a Nvidia GTX 295 dual GPU with 894 $GFlops$ processing power and 896 MBs memory per GPU core. Currently, the developed system utilizes a single GPU core.

Each workstation is connected to a camera pair. Cameras are synchronized by a timestamp-based software that utilizes a dedicated *FireWire 2* interface (800 $MBits/sec$) which guarantees a maximum of 125 μsec temporal discrepancy in images with the same timestamp. Eight images sharing the same timestamp constitute a *multiframe*.

2.2 Processing Pipeline

Cameras are extrinsically and intrinsically calibrated based on the method and tools reported in [7]. The processing pipeline consists of the CPU workflow, responsible for image acquisition and communication management and the GPU workflow, where the 3D reconstruction pipeline has been implemented. Both processes are detailed in the following.

CPU Workflow and Networking
Each workstation holds in its RAM a buffer of fixed size for every camera that



(a) (b)

Fig. 1. The developed platform (a) schematic diagram (b) actual configuration.

is connected to it. Each buffer stores the captured frames after they have been converted from Bayer Tile to RGB format. Moreover, prior to storing in buffer, each image is transformed so that geometric distortions are cancelled out based on the available calibration information. The rate of storing images into buffers matches the camera's acquisition frame rate. Image data are stored together with their associated timestamps. To avoid buffer overflow as newer frames arrive, older frames are removed.

Each time a new image enters a buffer in a satellite workstation, its timestamp is broadcasted to the central workstation. This way, at every time step the central workstation is aware of which frames are stored in the satellite buffers. The same is also true for central's local buffers. During the creation of a multiframe, the central workstation selects the appropriate timestamps for each buffer, local or remote. Then, it broadcasts timestamp queries to the satellite workstations and acquires as response the queried frames, while for local buffers it just fetches the frames from its main memory. The frame set that is created in this way constitutes the multiframe for the corresponding time step. The process is shown schematically in Fig. 2.

GPU Workflow

After a multiframe has been assembled, it is uploaded on the GPU for further processing. Initially, a pixel-wise parallelized foreground detection procedure is applied to the synchronized frames. The algorithm labels each pixel either as background or foreground, providing binary silhouette images as output. The produced silhouette set is given as input to a shape-from-silhouette 3D reconstruction process which, in turn, outputs voxel occupancy information. The occupancy data are then send to an instance of a parallel marching cubes algorithm for computing the surfaces of reconstructed objects. Optionally, prior to mesh

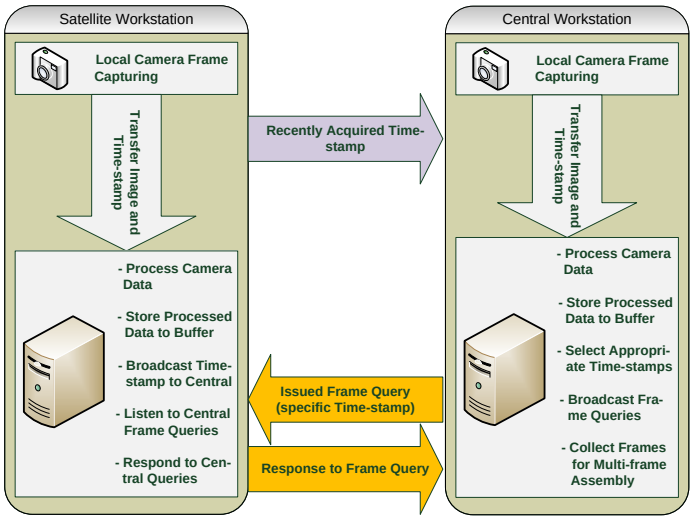


Fig. 2. Multiframe acquisition process.

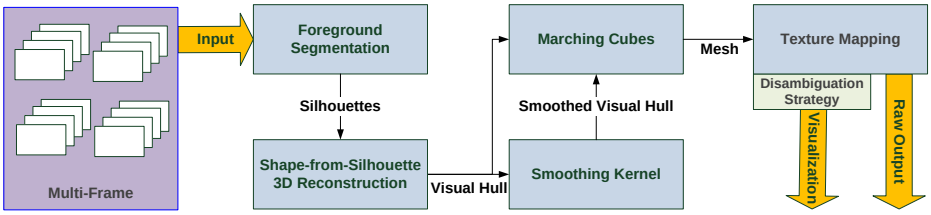


Fig. 3. GPU workflow.

calculation, the voxel representation is convolved with a 3D mean filter kernel to produce a smoothed output. Then, the texture of the original images is mapped onto the triangles of the resulted mesh. During this step multiple texture coordinate pairs are computed for each triangle. Each pair, projects the triangle's vertices at each view the triangle's front face is visible from. A disambiguation strategy is later incorporated to resolve the multi-texturing conflicts. Finally, results are formatted into appropriate data structures and returned to the CPU host program for further processing. In case the execution is intended for visualization, the process keeps the data on the GPU and returns to the host process handles to DirectX or OpenGL data structures (i.e. vertex and texture buffers). These are consequently used with proper graphics API manipulation for onscreen rendering. The overall procedure is presented schematically in Fig. 3.

3 GPU Implementation

In this section, the algorithms implemented on the GPU are presented in detail.

3.1 Foreground Segmentation

The terms *foreground segmentation* and *background subtraction* refer to methods that detect and segment moving objects in images captured by static cameras. Due to the significance and necessity of such methods a great number of approaches have been proposed. The majority of these approaches define pixel-wise operations [8]. The most straightforward of those subtract the average, median or running average within a certain time window from static views. Others utilize kernel density estimators and mean-shift based estimation [9, 10].

A very popular approach [11] that achieves great performance defines each image pixel's appearance model as a mixture of Gaussian components. This method is able to model complex background variations. Targeted at systems operating in relatively controlled environments (i.e., indoor environments with controlled lighting conditions) this work is based on the parallelization of the background modeling and foreground detection work of [12] which considers the appearance of a background pixel to be modeled by a single Gaussian distribution. This reduces substantially both the memory requirements and the overall computational complexity of the resulting process. Moreover, the assumption that pixels are independent, indicates the inherent parallelism of this algorithm. In addition, our implementation incorporates a technique for shadow detection that is also used in [13] and described thoroughly in [14]. Detected shadows are always labeled as background.

Formally, let $I^{(t)}$ correspond to an image of the multiframe acquired at timestamp t, and let $x^{(t)}$ be a pixel of this image represented in some colorspace. The background model is initialized by the first image of the sequence (i.e. $I^{(0)}$) and is given by

$$\hat{p}(\boldsymbol{x}|x^{(0)}, BG) = N(\boldsymbol{x}; \hat{\boldsymbol{\mu}}, \hat{\sigma}^2 I), \tag{1}$$

with $\hat{\boldsymbol{\mu}}$ and $\hat{\sigma}^2$ being the estimates of mean and variance of the Gaussian, respectively. In order to compensate for gradual global light variation, the estimations of $\boldsymbol{\mu}$ and σ are updated at every time step through the following equations:

$$\hat{\boldsymbol{\mu}}^{(t+1)} \leftarrow \hat{\boldsymbol{\mu}}^{(t)} + o^{(t)} \alpha_\mu \boldsymbol{\delta_\mu}^{(t)} \tag{2}$$

$$\sigma^{(t+1)} \leftarrow \sigma^{(t)} + o^{(t)} \alpha_\sigma \delta_\sigma{}^{(t)}, \tag{3}$$

where $\boldsymbol{\delta_\mu} = \boldsymbol{x}^{(t)} - \boldsymbol{\mu}^{(t)}$, $\delta_\sigma = |\boldsymbol{\mu}^{(t)} - \boldsymbol{x}^{(t)}|^2 - \sigma^{(t)}$ and a_μ, a_σ are the update factors for mean and standard deviation, respectively, and

$$o^{(t)} = \begin{cases} 1 & if\, x^{(t)} \in BG \\ 0 & if\, x^{(t)} \in FG. \end{cases} \tag{4}$$

A newly arrived sample is considered as background if the sample's distance to the background mode is less than four standard deviations. If this does not hold,

an additional condition is examined to determine whether the sample belongs to the foreground or it is a shadow on the background:

$$T_1 \leq \frac{\boldsymbol{\mu} \cdot \boldsymbol{x}^{(t)}}{|\boldsymbol{\mu}|^2} \leq 1 \quad and \quad \left| \left(\frac{\boldsymbol{\mu} \cdot \boldsymbol{x}^{(t)}}{|\boldsymbol{\mu}|^2} \right) \boldsymbol{\mu} - \boldsymbol{x} \right|^2 < \sigma^2 T_2 \left(\frac{\boldsymbol{\mu} \cdot \boldsymbol{x}^{(t)}}{|\boldsymbol{\mu}|^2} \right)^2, \quad (5)$$

where T_1, T_2, are empirically defined thresholds that are set to $T_1 = 0.25$, $T_2 = 150.0$.

The above described foreground detection method has been parallelized in a per pixel basis. In addition, because there is a need to preserve the background model for each view, this is stored and updated on GPU during the entire lifetime of the reconstruction process. In order to keep the memory requirements low and to meet the GPU alignment constrains, the background model of each pixel is stored in a 4-byte structure. This representation leads to a reduction of precision. Nevertheless, it has been verified experimentally that this does not affect noticeably the quality of the produced silhouettes.

3.2 Visual Hull Computation

The idea of *volume intersection* for the computation of a volumetric object description was introduced in the early 80's [15] and has been revisited in several subsequent works [16–18]. The term *visual hull*, is defined as the maximal shape that projects to the same silhouettes as the observed object on all views that lay outside the convex hull of the object [19].

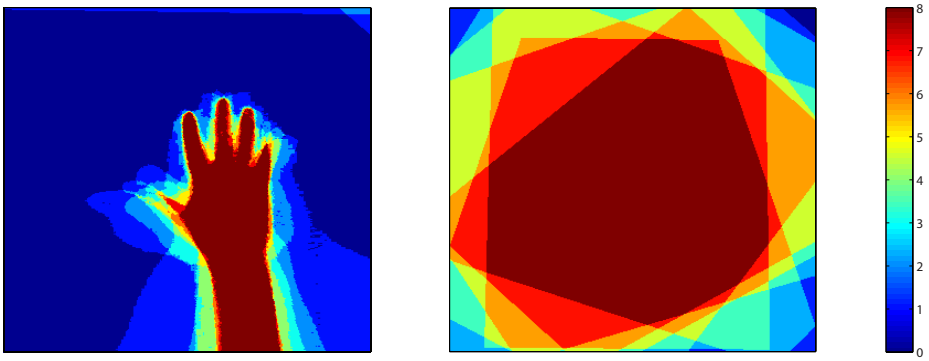
To compute the visual hull, every silhouette image acquired from a given multiframe, is back-projected and intersected into the common 3D space along with all others, resulting to the *inferred visual hull*, i.e. a voxel representation containing occupancy information. In this 3D space, a fixed size volume is defined and sampled to produce a 3D grid, $G = \left\{G^0, G^1, \ldots, G^n\right\}$ where $G^c = (X_c, Y_c, Z_c)$. Let C_i be the calibration matrix of camera i and R_i, T_i the corresponding rotation matrix and translation vector respectively, in relation to the global world-centered coordinate system. The general perspective projection of a point G expressed in homogeneous coordinates (i.e. $(X_c, Y_c, Z_c, 1)$) to the i^{th} view plane is described through the following equation

$$(x_c, y_c, f_c)^T = C_i \left[R_i | T_i\right] (X_c, Y_c, Z_c, 1)^T, \quad (6)$$

where $P_i = C_i \left[R_i | T_i\right]$ is the projection matrix of the corresponding view. Each point can be considered to be the mass center of some voxel on the defined 3D volume. We also define two additional functions. The first, labels projections falling inside the FOV of camera i as

$$L_i(x, y) = \begin{cases} 1 & 1 \leq x \leq w_i \ \wedge 1 \leq y \leq h_i \\ 0 & otherwise, \end{cases} \quad (7)$$

where w_i and h_i denote the width and height of the corresponding view plane, respectively. The second function measures the occupancy scores of each voxel



(a) (b)

Fig. 4. Each figure presents a xy plane slice of the voxel space. (a) The intersection of the projected silhouettes in slice $Z_{slice} = 90cm$. (b) The voxel space defined in this example is much larger than the previous, visibility factor variations are shown with different colors. Dark red denotes an area visible by all views.

via its projected center of mass, as

$$O(X_k, Y_k, Z_k) = \begin{cases} 1 & s = l > \frac{|C|}{2} \\ 0 & otherwise \end{cases}, \quad \forall k \in [0, n], \tag{8}$$

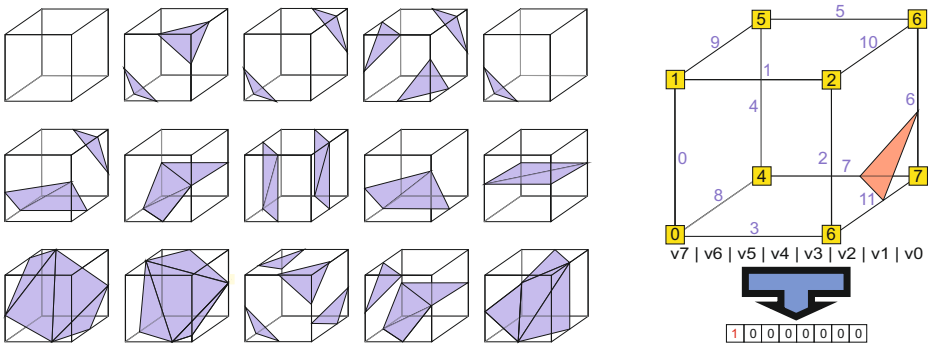
where $|C|$ is the number of views. l is the *visibility factor*, s the *intersection factor* and are defined as

$$l = \sum_{i \in C} L_i \left(\frac{x_k^i}{f_k^i}, \frac{y_k^i}{f_k^i} \right), \quad s = \sum_{i \in C} S_i \left(\frac{x_k^i}{f_k^i}, \frac{y_k^i}{f_k^i} \right), \tag{9}$$

with $\left(x_k^i/f_k^i, y_k^i/f_k^i\right)$ be the projection of (X_k, Y_k, Z_k) at view i and $S_i(x, y)$ is the function that takes value 1 if at view i the pixel (x, y) is a foreground pixel and 0 otherwise (i.e. background pixel). Figure 4 illustrates graphically the notion of l and s.

The output of the above process is the set $O(X_k, Y_k, Z_k)$ of occupancy values that represent the visual hull of the reconstructed objects. It can also be conceived as the estimation of a 3D density function. Optionally, the visual hull can be convolved with a 3D mean filter to smooth out the result. Due to its high computational requirements, this method targets the offline mode of 3D reconstruction.

The above described 3D reconstruction process has been parallelized on a per 3D point basis. More specifically, each grid point is assigned to a single GPU thread responsible for executing the above mentioned calculations. To speed up the process, shared memory is utilized for storing the static per thread block calibration information, silhouette images are preserved in GPU texture memory in a compact bit-per-pixel format and occupancy scores are mapped to single bytes.



(a) (b)

Fig. 5. (a) Marching Cubes fundamental states. (b) Byte representation and indexing.

3.3 Marching Cubes

Marching cubes [20, 21] is a popular algorithm for calculating isosurface descriptions out of density function estimates. Due to its inherent and massive data parallelism it is ideal for GPU implementation. Over the last few years, a lot of isosurface calculation variates that utilize GPUs have been proposed [22–26]. In this work we employ a slightly modified version of the marching cubes implementation found at [27] due to its simplicity and speed. More specifically, the occupancy grid resulting from 3D visual hull estimation is mapped into a CUDA 3D texture. Each voxel is assigned to a GPU thread. During calculations, each thread samples the density function (i.e. CUDA 3D texture) at the vertices of it's corresponding voxel. The normalized (in the range $[0, 1]$), bilinearly interpolated, single precision values returned by this step, represent whether the voxel vertices are located inside or outside a certain volume. We consider the isosurface level to be at 0.5. Values between 0 and 1, also show how close a voxel vertex is to the isosurface level. Using this information, a voxel can be described by a single byte, where each bit corresponds to a vertex and is set to 1 or to 0 if this vertex lays inside or outside a volume, respectively. There are 256 discrete generic states in which a voxel can be intersected by an isosurface fragment, produced from the 15 fundamental cases illustrated in Fig. 5a.

Parallel marching cubes uses two constant lookup tables for its operation. The first lookup table is indexed by the voxel byte representation and is utilized for determining the number of triangles the intersecting surface consists of. The second table is a 2D array, where its first dimension is indexed by the byte descriptor and the second by an additional index $trI \in [0, 3N_{iso} - 1]$ where N_{iso} is the number of triangles returned by the first lookup. Given a byte index, sequential triplets accessed through successive trI values, contain the indices of voxel vertices intersected by a single surface triangle. An example of how the voxel byte descriptor is formed is shown in Fig. 5b. This figure also presents the

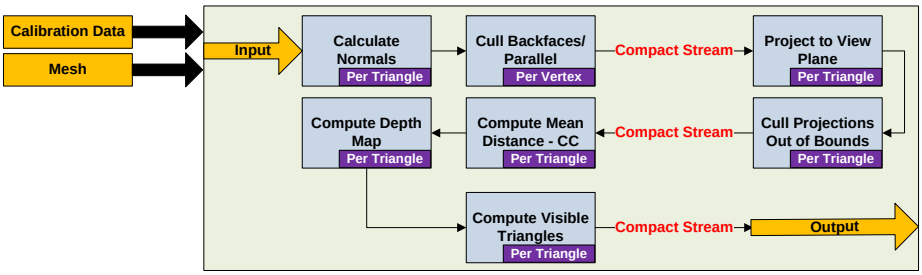


Fig. 6. Computation of texture coordinates.

vertex and edge indexing along with an example of an intersecting isosurface fragment that consists of a single triangle.

To avoid applying this process to all voxels, our implementation determines the voxels that are intersected by the iso-surface and then, using the CUDA data parallel primitives library [28], applies stream compaction through the exclusive sum-scan algorithm [29] to produce the minimal voxel set containing only intersected voxels. Finally, lookup tables are mapped to texture memory for fast access.

3.4 Texture Mapping

Due to the fact that the employed camera setup provides multiple texture sources, texture mapping of a single triangle can be seen as a three step procedure: a) determine the views from which the triangle is visible, b) compute the corresponding texture coordinates and c) apply a policy for resolving multitexturing ambiguities (i.e. texture selection). The current implementation carries out the first two steps in a per view manner i.e.: a) determines the subset of triangles that are visible by a certain view and b) computes their projections on view plane. The third step is applied either on a per pixel basis through a pixel shader during the visualization stage, or is explicitly computed by the consumer of the offline dataset.

Specifically, given the calibration data for a view and the reconstructed mesh, a first step is the calculation of the triangle normals. Then, the direction of each camera's principal axis vector is used to cull triangles back-facing the camera or having an orientation (near-)parallel to the camera's view plane. The triangle stream is compacted excluding culled polygons and the process continues by computing the view plane projections of the remaining triangles. Projections falling outside the plane's bounds are also removed through culling and stream compaction. Subsequently, the mean vertex distance from the camera center is computed for each remaining triangle and a depth testing procedure (Z-buffering) is applied to determine the final triangle set. The procedure is shown schematically in Fig. 6. This figure also shows the granularity of the decomposition in independent GPU threads. During depth testing, CUDA atomics are used for issuing writes on the depth map. The reason for the multiple culling

Image resolution	Multiframe acquisition	Foreground segmentation
320×240	30 *mfps*	$22.566,3$ *fps* / $2.820,8$ *mfps*
640×480	13 *mfps*	$6.773,9$ *fps* / $846,4$ *mfps*
800×600	9 *mfps*	$4.282,6$ *fps* / $535,3$ *mfps*
1280×960	$3,3$ *mfps*	$1.809,9$ *fps* / $226,2$ *mfps*

Table 1. Performance of acquisition and segmentation for various image resolutions.

iterations prior to depth testing is for keeping the thread execution queues length minimal during serialization of depth map writes.

There is a number of approaches that one can use to resolve multitexturing conflicts. Two different strategies have been implemented in this work. The first assigns to each triangle the texture source at which the projection area is maximal among all projections. The second blends all textures according to a weighting factor, proportional to the size of the projected area. A special case is the one where all weights are equal. This last approach is used during online experiments to avoid the additional overhead of computing and comparing the projection areas, while the others are used in offline mode for producing better quality results. In online mode the process is applied through a pixel shader implemented using HLSL and shader model 3.0. Visualizations of a resulted mesh are shown in Fig. 7. The supplemental material attached to this paper shows representative videos obtained from both online and offline experiments.

4 Performance

Given a fixed number of cameras, the overall performance is determined by the network bandwidth, the size of transferred data, the GPU execution time and the quality of the reconstruction. In online experiments, camera images are preprocessed, transferred through network and finally collected at the central workstation to construct a synchronized multiframe. This is performed at a rate of 30 multiframes per second (mfps). To achieve this performance, original images (i.e. 1280×960) are resized during the CPU preprocessing stage to a size of 320×240. Further reduction of image resolution increases the framerate beyond real-time (i.e. $\geq 30\, mfps$) at the cost of reducing the 3D reconstruction quality. Table 1 shows the achieved multiframe acquisition speed.

Table 1 also shows that, as expected, foreground segmentation speed is linearly proportional to image size. These last reported measurements do not include CPU/GPU memory transfers.

The number of voxels that constitute the voxel space is the primary factor that affects the quality of the reconstruction and overall performance. Given a bounded voxel space (i.e., observed volume), smaller voxel sizes, produce more accurate estimates of the 3D density function leading to a reconstruction output of higher accuracy. Moreover, higher voxel space resolutions issue greater numbers of GPU threads and produce more triangles for the isosurface that, in turn, leads to an increased overhead during texture mapping. The performance

(a) (b) (c)

Fig. 7. 3D reconstruction of a single multiframe: (a) no smoothing, (b) smoothed reconstruction and (c) smoothed and textured output.

graph of Fig. 8a shows the overall performance impact of voxel space resolution increment in the cases of a) no smoothing of the visual hull, b) smoothed hull utilizing a 3^3 kernel and c) smoothed hull utilizing a 5^3 kernel. The graph in Fig. 8b presents computational performance as a function of smoothing kernel size. In both graphs, multiframe processing rate corresponds at the processing rate of the entire GPU pipeline including the CPU/GPU memory transfer times. It is worth mentioning that although image resolution affects the quality of the reconstruction and introduces additional computational costs due to the increased memory transfer and foreground segmentation overheads, it does not have a significant impact on the performance of the rest of the GPU reconstruction pipeline.

Table 2 presents quantitative performance results obtained from executed experiments. In the 3rd and 4th columns, the performance of 3D reconstruction and texture mapping are shown independently. The 3D reconstruction column corresponds to the processes of computing the visual hull, smoothing the occupancy volume and creating the mesh, while texture mapping column corresponds to the

Voxels	**Smoothing**	**3D reconst.**	**Text. mapping**	**Output**
Online Experiments				
$120 \times 140 \times 70$	No	$136,8$ *mfps*	$178,0$ *mfps*	$64,0$ *mfps*
$100 \times 116 \times 58$	No	$220,5$ *mfps*	$209,9$ *mfps*	$84,5$ *mfps*
Offline Experiments				
$277 \times 244 \times 222$	Kernel: 3^3	$7,7$ *mfps*	$27,5$ *mfps*	$5,0$ *mfps*
$277 \times 244 \times 222$	Kernel : 5^3	$4,7$ *mfps*	$28,9$ *mfps*	$3,5$ *mfps*
$277 \times 244 \times 222$	No	$11,4$ *mfps*	$25,3$ *mfps*	$6,2$ *mfps*

Table 2. Quantitative performance results obtained from representative experiments. Image resolution is set to 320×240 for online and 1280×960 for offline experiments.

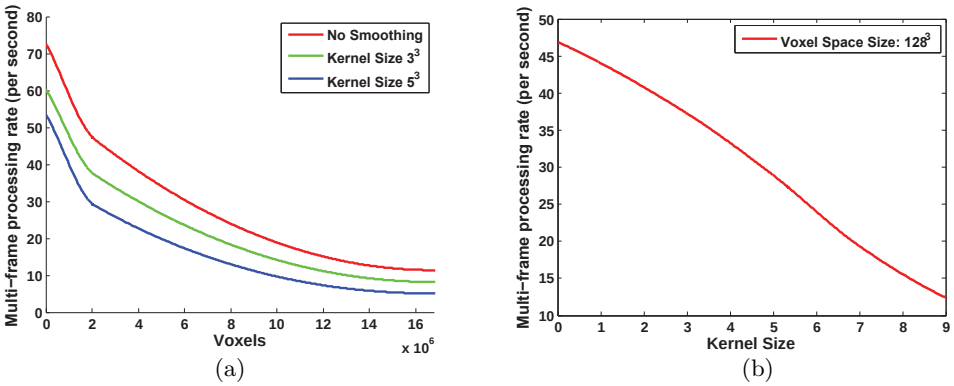


Fig. 8. Performance graphs. Image resolution is set to 640 × 480 in all experiments. (a) Performance impact of voxel space descretization resolution. (b) The performance effect of 3D smoothing kernel size.

performance of the process depicted in Fig. 6. Finally, in the output column, as in the previous experiments, the performance of the entire reconstruction pipeline is measured including foreground segmentation and memory transfers. It can be seen that keeping the voxel space resolution at a fixed size, the multiframe processing rate of 3D reconstruction drops significantly when the smoothing process is activated. On the contrary, texture mapping is actually accelerated due to the fact that the smoothed surface is described by less triangles than the original one. Online experiments present clearly the effect of the voxel space resolution in overall performance.

5 Conclusions - Future Work

In this paper, we presented the design and implementation of an integrated, GPU-powered, multicamera vision system that is capable of performing foreground image segmentation, silhouette-based 3D reconstruction, 3D mesh computation and texture mapping in real-time. In online mode, the developed system can support higher level processes that are responsible for activity monitoring and interpretation. In offline mode, it enables the acquisition of high quality 3D datasets. Experimental results provide a quantitative assessment of the system's performance. Additionally, the supplementary material provides qualitative evidence regarding the quality of the obtained results.

The current implementation utilizes a single GPU. A future work direction is the incorporation of more GPUs either on central or satellite workstations, to increase the system's overall raw computational power in terms of GFlops. In this case, an intelligent method for distributing the computations over the entire GPU set must be adopted, while various difficult concurrency and synchronization issues that this approach raises must be addressed. Furthermore,

performance gains could be achieved by transferring the image post-acquisition CPU processes of Bayer Tile-to-RGB conversion and distortion correction to GPUs as they also encompass a high degree of parallelism. Finally, mesh deformation techniques instead of density function smoothing and advanced texture source disambiguation/blending strategies that incorporate additional information (e.g. edges) can be utilized in order to further augment the quality of the results.

Acknowledgments

This work was partially supported by the IST-FP7-IP-215821 project GRASP and by the FORTH-ICS internal RTD Programme "Ambient Intelligence and Smart Environments".

References

1. Kim, H., Sakamoto, R., Kitahara, I., Toriyama, T., Kogure, K.: Compensated visual hull with gpu-based optimization. Advances in Multimedia Information Processing-PCM 2008 (2008) 573–582
2. Schick, A., Stiefelhagen, R.: Real-time gpu-based voxel carving with systematic occlusion handling. In: Pattern Recognition: 31st DAGM Symposium, Jena, Germany, September 9-11, 2009, Proceedings, Springer-Verlag New York Inc (2009) 372
3. Matusik, W., Buehler, C., Raskar, R., Gortler, S.J., McMillan, L.: Image-based visual hulls. In: SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques, New York, NY, USA, ACM Press/Addison-Wesley Publishing Co. (2000) 369–374
4. Matsuyama, T., Wu, X., Takai, T., Nobuhara, S.: Real-time 3d shape reconstruction, dynamic 3d mesh deformation, and high fidelity visualization for 3d video. Computer Vision and Image Understanding **96** (2004) 393–434
5. Ladikos, A., Benhimane, S., Navab, N.: Efficient visual hull computation for real-time 3d reconstruction using cuda. In: IEEE Conference on Computer Vision and Pattern Recognition, Workshops 2008. (2008) 1–8
6. Waizenegger, W., Feldmann, I., Eisert, P., Kauff, P.: Parallel high resolution real-time visual hull on gpu. In: IEEE International Conference on Image Processing. (2009) 4301–4304
7. Sarmis, T., Zabulis, X., Argyros, A.A.: A checkerboard detection utility for intrinsic and extrinsic camera cluster calibration. Technical Report TR-397, FORTH-ICS (2009)
8. Piccardi, M.: Background subtraction techniques: a review. In: IEEE International Conference on Systems, Man and Cybernetics. Volume 4. (2004) 3099–3104
9. Elgammal, A., Harwod, D., Davis, L.: Non-parametric model for background subtraction. In: IEEE International Conference on Computer Vision, Frame-rate Workshop. (1999)
10. Han, B., Comaniciu, D., Davis, L.: Sequential kernel density approximation through mode propagation: applications to background modeling. In: Asian Conference on Computer Vision. (2004)

11. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: IEEE Conference on Computer Vision and Pattern Recognition. (1999) 246–252
12. Wren, C., Azarbayejani, A., Darrell, T., Pentland, A.: Pfinder: Real-time tracking of the human body. IEEE Transactions on Pattern Analysis and Machine Intelligence **19** (1997) 780–785
13. Zivkovic, Z.: Improved adaptive gaussian mixture model for background subtraction. In: International Conference on Pattern Recognition. (2004)
14. Prati, A., Mikic, I., Trivedi, M.M., Cucchiara, R.: Detecting moving shadows: Algorithms and evaluation. IEEE Transactions on Pattern Analysis and Machine Intelligence **25** (2003) 918–923
15. Martin, W., Aggrawal, J.: Volumetric descriptions of objects from multiple views. IEEE Transactions on Pattern Analysis and Machine Intelligence (1983)
16. Srinivasan, P., Liang, P., Hackwood, S.: Computational geometric methods in volumetric intersection for 3d reconstruction. Pattern Recognition **23** (1990) 843 – 857
17. Greg, F.P., Slabaugh, G., Culbertson, B., Schafer, R., Malzbender, T.: A survey of methods for volumetric scene reconstruction. In: International Workshop on Volume Graphics. (2001)
18. Potmesil, M.: Generating octree models of 3d objects from their silhouettes in a sequence of images. Computer Vision, Graphics, and Image Processing **40** (1987) 1–29
19. Laurentini, A.: The visual hull concept for silhouette-based image understanding. IEEE Transactions on Pattern Analysis and Machine Intelligence **16** (1994) 150–162
20. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. Computer Graphics **21** (1987) 163–169
21. Newman, T.S., Yi, H.: A survey of the marching cubes algorithm. Computers and Graphics **30** (2006) 854– 879
22. Klein, T., Stegmaier, S., Ertl, T.: Hardware-accelerated reconstruction of polygonal isosurface representations on unstructured grids. In: PG '04: Proceedings of the Computer Graphics and Applications, 12th Pacific Conference, Washington, DC, USA, IEEE Computer Society (2004) 186–195
23. Pascucci, V.: Isosurface computation made simple: Hardware acceleration, adaptive refinement and tetrahedral stripping. In: In Joint Eurographics - IEEE TVCG Symposium on Visualization (VisSym. (2004) 293–300
24. Reck, F., Dachsbacher, C., Grosso, R., Greiner, G., Stamminger, M.: Realtime isosurface extraction with graphics hardware. In: Proceedings of Eurographics. (2004)
25. Goetz, F., Junklewitz, T., Domik, G.: Real-time marching cubes on the vertex shader. In: Proceedings of Eurographics. Volume 2005. (2005)
26. Johansson, G., Carr, H.: Accelerating marching cubes with graphics hardware. In: In CASCON 06: Proceedings of the 2006 conference of the Center for Advanced Studies on Collaborative research, ACM, Press (2006) 378
27. NVIDIA. GPU Computing SDK (2009) http://developer.nvidia.com/object/gpu computing.html.
28. Harris, M., Sengupta, S., Owens, J. CUDA Data Parallel Primitives Library (2007) http://code.google.com/p/cudpp/.
29. Sengupta, S., Harris, M., Zhang, Y., Owens, J.D.: Scan primitives for gpu computing. In: Graphics Hardware 2007, ACM (2007) 97–106

Object Tracking and Segmentation in a Closed Loop

Konstantinos E. Papoutsakis and Antonis A. Argyros

Institute of Computer Science, FORTH
and
Computer Science Department, University of Crete
{papoutsa,argyros}@ics.forth.gr
http://www.ics.forth.gr/cvrl/

Abstract. We introduce a new method for integrated tracking and segmentation of a single non-rigid object in an monocular video, captured by a possibly moving camera. A closed-loop interaction between EM-like color-histogram-based tracking and Random Walker-based image segmentation is proposed, which results in reduced tracking drifts and in fine object segmentation. More specifically, pixel-wise spatial and color image cues are fused using Bayesian inference to guide object segmentation. The spatial properties and the appearance of the segmented objects are exploited to initialize the tracking algorithm in the next step, closing the loop between tracking and segmentation. As confirmed by experimental results on a variety of image sequences, the proposed approach efficiently tracks and segments previously unseen objects of varying appearance and shape, under challenging environmental conditions.

1 Introduction

The vision-based tracking and the segmentation of an object of interest in an image sequence are two challenging computer vision problems. Each of them has its own importance and challenges and can be considered as "chicken-and-egg" problems. By solving the segmentation problem, a solution to the tracking problem can easily be obtained. At the same time, tracking provides important input to segmentation.

In a recent and thorough review on the state-of-the-art tracking techniques [1], tracking methods are divided into three categories: *point tracking*, *silhouette tracking* and *kernel tracking*. Silhouette-based tracking methods usually evolve an initial contour to its new position in the current frame. This can be done using a state space model [2] defined in terms of shape and motion parameters [3] of the contour or by the minimization of a contour-based energy function [4, 5], providing an accurate representation of the tracked object. Point-tracking algorithms [6, 7] can also combine tracking and fine object segmentation using multiple image cues. Towards a more reliable and drift-free tracking, some point tracking algorithms utilize energy minimization techniques, such as Graph-Cuts or Belief Propagation on a Markov Random Field (MRF) [8] or on a Conditional

G. Bebis et al. (Eds.): ISVC 2010, Part I, LNCS 6453, pp. 405–416, 2010.

Random Field (CRF) [9, 10]. Most of the kernel-based tracking algorithms [11–13] provide a coarse representation of the tracked object based on a bounding box or an ellipsoid region.

Despite the many important research efforts devoted to the problem, the development of algorithms for tracking objects in unconstrained videos constitutes an open research problem. Moving cameras, appearance and shape variability of the tracked objects, varying illumination conditions and cluttered backgrounds constitute some of the challenges that a robust tracking algorithm needs to cope with. To this end, in this work we consider the combined tracking and segmentation of previously unseen objects in monocular videos captured by a possibly moving camera. No strong constraints are imposed regarding the appearance and the texture of the target object or the rigidity of its shape. All of the above may dynamically vary over time under challenging illumination conditions and changing background appearance. The basic aim of this work is to preclude tracking failures by enhancing its target localization performance through fine object segmentation that is appropriately integrated with tracking in a closed-loop algorithmic scheme. A kernel-based tracking algorithm [14], a natural extension of the popular mean-shift tracker [11, 15], is efficiently combined with Random Walker-based image segmentation [16, 17]. Explicit segmentation of the target region of interest in an image sequence enables reliable tracking and reduces drifting by exploiting static image cues and temporal coherence.

The key benefits of the proposed method are (i) close-loop interaction between tracking and segmentation (ii) enhanced tracking performance under challenging conditions (iii) fine object segmentation (iv) capability to track objects from a moving camera (v) increased tolerance to extensive changes of object's appearance and shape and, (vi) continual refinement of both the object and the background appearance models.

The rest of the paper is organized as follows. The proposed method is presented in Sec. 2. Experimental results are presented in Sec. 3. Finally, Sec. 4 summarizes the main conclusions from this work and future work perspectives.

2 Proposed Method

For each input video frame, the proposed framework encompasses a number of algorithmic steps, tightly interconnected in a closed-loop which is illustrated schematically in Fig.1. To further ease understanding, Fig.2 provides sample intermediate results of the most important algorithmic steps.

The method assumes that at a certain moment t in time, a new image frame I_t becomes available and that a fine object segmentation mask M_{t-1} is available as the result of the previous time step $t-1$. For time $t=0$, M_{t-1} should be provided for initialization purposes. Essentially, M_{t-1} is a binary image where foreground/background pixels have a value of 1/0, respectively (see Fig.2). The goal of the method is to produce the current object segmentation mask M_t. Towards this end, the spatial mean and covariance matrix of the foreground region of M_{t-1} is computed, thus defining an ellipsoid region coarsely representing

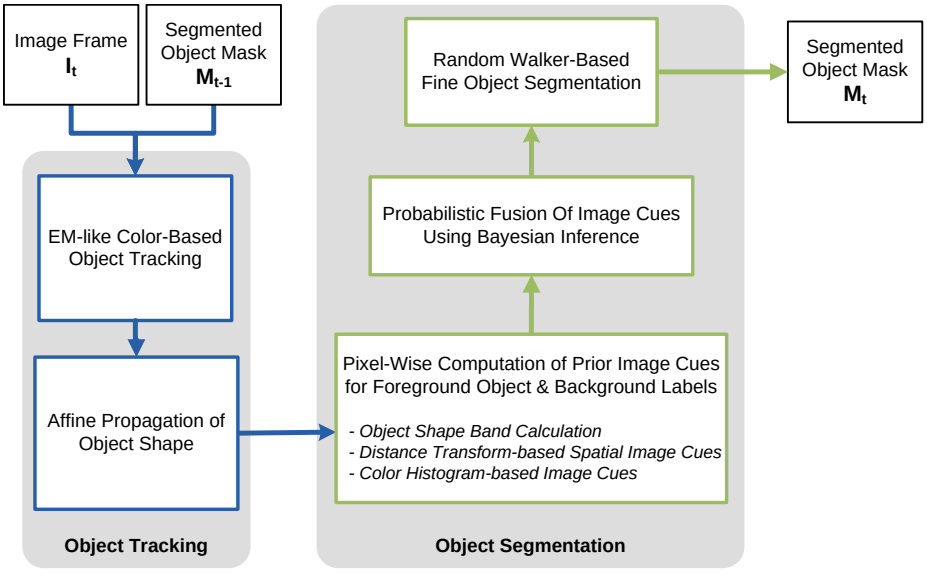


Fig. 1. Outline of the proposed method

the object at $t-1$. Additionally, a color-histogram-based appearance model of the segmented object (i.e., the one corresponding to the foreground of M_{t-1}) is computed using a Gaussian weighting kernel function. The iterative (EM-like) tracking algorithm in [14] is initialized based on the computed ellipsoid and appearance models. The tracking thus performed, results in a prediction of the position and covariance of the ellipsoid representing the tracked object. Based on the transformation parameters of the ellipsoid between $t-1$ and t, a 2D spatial affine transformation of the foreground object mask M_{t-1} is performed. The propagated object mask $M_t^{'}$ indicates the predicted position and shape of the object in the new frame I_t. The Hausdorff distance between the contour points of M_{t-1} and $M_t^{'}$ masks is then computed and a *shape band*, as in [4, 9], around the $M_t^{'}$ contour points is determined, denoted as B_t. The width of B_t is equal to the computed Hausdorff distance of the two contours. This is performed to guarantee that the shape band contains the actual contour pixels of the tracked object in the new frame. Additionally, the pixel-wise Distance Transform likelihoods for the object and background areas are computed together with the pixel-wise color likelihoods based on region-based color histograms. Pixel-wise Bayesian inference is applied to fuse spatial and color image cues, in order to compute the probability distribution for the object and the background regions. Given the estimated Probability Density Functions (PDFs) for each region, a Random Walker-based segmentation algorithm is finally employed to obtain M_t in I_t.

In the following sections, the components of the proposed method are described in more detail.

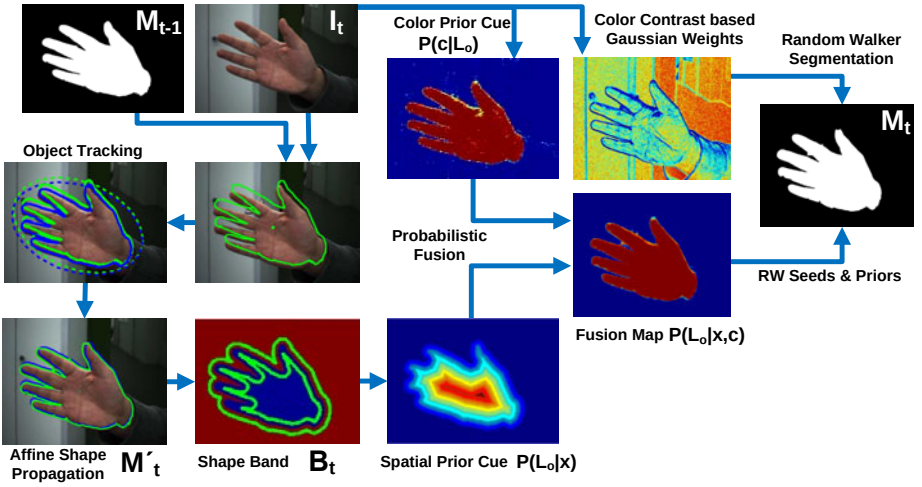


Fig. 2. Sample intermediate results of the proposed method. To avoid clutter, results related to the processing of the scene background are omitted.

2.1 Object Tracking

This section presents the tracking part of the proposed combined tracking and segmentation method (see the bottom-left part of Fig.1).

EM-Like Color Based Object Tracking: The tracking method [14] used in this work is closely related to the widely-used mean-shift tracking method [11, 15]. More specifically, this algorithm coarsely represents the objects' shape by a 2D ellipsoid region, modeled by its center $\boldsymbol{\theta}$ and covariance matrix Σ. The appearance model of the tracked object is represented by the color histogram of the image pixels under the 2D ellipsoid region corresponding to $\boldsymbol{\theta}$ and Σ, and is computed using a Gaussian weighting kernel function. Provided M_{t-1} and I_{t-1}, $\boldsymbol{\theta}_{t-1}$, Σ_{t-1} the object appearance model can be computed for time $t-1$. Given a new image frame I_t where the tracked object is to be localized, the tracking algorithm evolves the ellipsoid region in order to determine the image area in I_t that best matches the appearance of the tracked object in terms of a Bhattacharrya coefficient-based color similarity measure. This gives rise to the parameters $\boldsymbol{\theta_t}$ and Σ_t that represent the predicted object position and covariance in I_t.

Affine Propagation of Object Shape: The tracking algorithm presented above assumes that the shape of an object can be accurately represented as an ellipse. In the general case, this is a quite limiting assumption, therefore the objects' appearance model is forced to include background pixels, causing tracking to drift. The goal of this work is to prevent tracking drifts by integrating tracking with fine object segmentation.

To accomplish that, the contour C_{t-1} of the object mask in M_{t-1} is propagated to the current frame I_t based on the transformation suggested by the parameters $\boldsymbol{\theta_{t-1}}$, $\boldsymbol{\theta_t}$, Σ_{t-1} and Σ_t. A 2D spatial, affine transformation is defined between the corresponding ellipses. Exploiting the obtained Σ_{t-1} and Σ_t covariance matrices, a linear 2×2 affine transformation matrix A_t can be computed based on the square root ($\Sigma^{1/2}$) of each of these matrices. It is known that a covariance matrix is a square, symmetric and positive semidefinite matrix. The square root of any 2×2 covariance matrix Σ can be calculated by diagonalization as

$$\Sigma^{1/2} = Q\Lambda^{1/2}Q^{-1}, \tag{1}$$

where Q is the square 2×2 matrix whose i^{th} column is the eigenvector q_i of Σ and $\Lambda^{1/2}$ is the diagonal matrix whose diagonal elements are the square values of the corresponding eigenvalues. Since Σ is a covariance matrix, the inverse of its Q matrix is equal to the transposed matrix Q^T, therefore $\Sigma^{1/2} = Q\Lambda^{1/2}Q^T$. Accordingly, we compute the transformation matrix A_t by:

$$A_t = Q_t\Lambda_t^{1/2}\Lambda_{t-1}^{-1/2}Q_{t-1}^T. \tag{2}$$

Finally, C_t' is derived from C_t based on the following transformation

$$C_t' = A_t(C_t - \boldsymbol{\theta}_{t-1}) + \boldsymbol{\theta}_t. \tag{3}$$

The result indicates a propagated contour C_t', practically a propagated object mask $M_t^{'}$ that serves as a prediction of the position and the shape of the tracked object in the new frame I_t.

2.2 Object Segmentation

This section presents how the pixel-wise posterior values on spatial and color image cues are computed and fused using Bayesian inference in order to guide the segmentation of the tracked foreground object (see the right part of Fig.1).

Object Shape Band: An object shape band B_t is determined around the predicted object contour C_t'. Our notion of shape band is similar to those used in [4, 9]. B_t can be regarded as an area of uncertainty, where the true object contour lies in image I_t. The width of B_t is determined by the Euclidean, 2D Hausdorff distance between contours C_{t-1} and $C_t^{'}$ regarded as two point sets.

Spatial Image Cues: We use the Euclidean 2D Distance Transform to compute the probability of a pixel $\boldsymbol{x_i}$ in image I_t to belong to either the object L_o or the background L_b region, based on its 2D location $\boldsymbol{x_i} = (x, y)$ on the image plane. As a first step, the shape band B_t of the propagated object contour C_t' is considered and its inner and outer contours are extracted. The Distance Transform is then computed starting from the outer contour of B_t towards the inner part of the object. The probability $P(L_o|x_i)$ of a pixel to belong to the

object given its image location is set proportional to its normalized distance from the outer contour of the shape band. For pixels that lie outside the outer contour of B_t, it holds that $P(L_o|x_i) = \epsilon$, where ϵ is a small constant.

Similarly, we compute the Euclidean Distance Transform measure starting from the inner contour of B_t towards the exterior part of the object. The probability $P(L_b|x_i)$ of a pixel to belong to the background given its image location is set proportional to its normalized distance from the inner contour of the shape band. For pixels that lie inside the inner contour of B_t, it holds that $P(L_b|x_i) = \epsilon$.

Color Based Image Cues: Based on the segmentation M_{t-1} of the image frame I_{t-1}, we define a partition of image pixels Ω into sets Ω_o and Ω_b indicating the object and background image pixels, respectively. The appearance model of the tracked object is represented by the color histogram H_o computed on the Ω_o set of pixels. The normalized value in a histogram bin c encodes the conditional probability $P(c|L_o)$. Similarly, the appearance model of the background region is represented by the color histogram H_b, computed over pixels in Ω_b and encoding the conditional probability $P(c|L_b)$.

Probabilistic Fusion of Image Cues: Image segmentation can be considered as a pixel-wise classification problem for a number of classes/labels. Our goal is to generate the posterior probability distribution for each of the labels L_o and L_b, which will be further utilized to guide the Random Walker-based image segmentation. Using Bayesian inference, we formulate a probabilistic framework to efficiently fuse the available prior image cues, based on the pixel color and position information, as described earlier. Considering the pixel color c as the evidence and conditioning on pixel position x_i in image frame I_t, the posterior probability distribution for label L_l is given by

$$P(L_l \mid c, x_i) = \frac{P(c \mid L_l, x_i)P(L_l \mid x_i)}{\sum_{l=0}^{N} P(c \mid L_l, x_i)P(L_l \mid x_i)}, \tag{4}$$

where $N = 2$ in our case. The probability distribution $P(c \mid L_l, x_i)$ encodes the conditional probability of color c taking the pixel label L_l as the evidence and conditioning on its location x_i. We assume that knowing the pixel position x_i, does not affect our belief about its color c. Thus, the probability of color c is only conditioned on the prior knowledge of its class L_l following that $P(c \mid L_l, x_i) = P(c \mid L_l)$. Given this, Eq.(4) transforms to

$$P(L_l \mid c, x_i) = \frac{P(c \mid L_l)P(L_l \mid x_i)}{\sum_{l=0}^{N} P(c \mid L_l)P(L_l \mid x_i)}. \tag{5}$$

The conditional color probability $P(c \mid L_l)$ for the class L_l is obtained by the color histogram H_l. The conditional spatial probability $P(L_l \mid x_i)$ is obtained by the Distance-Transform measure calculation. Both of these calculations have been presented earlier in this section.

Random Walker Based Object Fine Segmentation: The resulting posterior distribution $P(L_l \mid c, x_i)$ for each of the two labels L_o and L_b on pixels x_i guides the Random Walker-based image segmentation towards an explicit and accurate segmentation of the tracked object in I_t.

Random Walks for image segmentation was introduced in [18] as a method to perform K-way graph-based image segmentation given a number of pixels with user (or automatically) defined labels, indicating the K disjoint regions in a new image that is to be segmented. The principal idea behind the method is that one can analytically determine the real-valued probability that a random walker starting at each unlabeled image pixel will first reach one of the pre-labeled pixels. The random walker-based framework bears some resemblance to the popular graph-cuts framework for image segmentation, as they are both related to the spectral clustering family of algorithms [19], but they also exhibit significant differences concerning their properties, as described in [17].

The algorithm is formulated on a discrete weighted undirected graph $G = (V, E)$, where nodes $u \in V$ represent the image pixels and the positive-weighted edges $e \in E \subseteq V \times V$ indicate their local connectivity. The solution is calculated analytically by solving *K-1* sparse, symmetric, positive-definite linear systems of equations, for K labels. For each graph node, the resulting probabilities of the potential labels sum up to 1.

In order to represent the image structure by random walker biases, we map the edge weights to positive weighting scores computed by the Gaussian weighting function on the normalized Euclidean distance of the color intensities between two adjacent pixels, practically the color contrast. The Gaussian weighting function is

$$w_{i,j} = e^{-\frac{\beta}{\rho}(\|c_i - c_j\|)^2} + \epsilon, \tag{6}$$

where c_i stands for the vector containing the color channel values of pixel/node i, ϵ is a small constant (i.e $\epsilon = 10^{-6}$) and ρ is a normalizing scalar $\rho = \max(\|c_i - c_j\|), \forall i, j \in E$. The parameter β is user-defined and modulates the spatial random walker biases, in terms of image edgeness. The posterior probability distribution $P(L_l \mid c, x_i)$ computed over the pixels x_i of the current image I_t suggest the probability of the pixels to be assigned to the label L_l. Therefore, we consider the pixels of highest posterior probability values for the label L_l as pre-labeled/seeds nodes of that label in the formulated graph.

An alternative formulation of the Random Walker-based image segmentation method is presented in [16]. This method incorporates non-parametric probability models, that is, prior belief on label assignments. In [16], the sparse linear systems of equations that need to be solved to obtain a real-valued density-based multilabel image segmentation are also presented. The two modalities of this alternative formulation suggest for using only prior knowledge on the belief of a graph node toward each of the potential labels, or using prior knowledge in conjunction with pre-labeled/seed graph nodes. The γ scalar weight parameter is introduced in these formulations, controlling the degree of effectiveness of the prior belief values towards the belief information obtained by the random walks. This extended formulation of using both seeds and prior beliefs on graph nodes

is compatible with our approach considering the obtained posterior probability distributions $P(L_l \mid c, x_i)$ for the two segmentation labels. The two Random Walker formulations that use prior models, suggest for a graph construction similar to the graph-cut algorithm [20], where the edge weights of the constructed graph can be seen as the *N-links* or *link-terms* and the prior belief values of the graph nodes for any of the potential labels can be considered as the *T-links* or the *data-terms*, in graph cuts terminology.

Regardless of the exact formulation used, the primary output of the algorithm consists of K probability maps, that is a soft image segmentation per label. By assigning each pixel to the label for which the greatest probability is calculated, a K-way segmentation is obtained. This process gives rise to object mask M_t for image frame I_t.

3 Experimental Results and Implementation Issues

The proposed method was extensively tested on a variety of image sequences. Due to space limitations, results on eight representative image sequences are presented in this paper. The objects tracked in these sequences go through extensive appearance, shape and pose changes. Additionally, these sequences differ with respect to the camera motion and to the lighting conditions during image acquisition which affects the appearance of the tracked objects.

We compare the proposed joint tracking and segmentation method with the tracking-only approach of [14]. The parameters of this algorithm were kept identical in the stand-alone run and in the run within the proposed framework. It is important to note that stand-alone tracking based on [14] is initialized with the appearance model extracted in the first frame of the sequence and that this appearance model is not updated over time. This is done because in all the challenging sequences we used as the basis of our evaluation, updating the appearance model based on the results of tracking, soon causes tracking drifts and total loss of the tracked object.

Figure 3 illustrates representative tracking results (i.e., five frames for each of the eight sequences). In the first sequence, a human hand undergoes complex articulations, whereas the lighting conditions significantly affect its skin color tone. In the second sequence, a human head is tracked despite its abrupt scale changes and the lighting variations. In the third sequence the articulations of a human hand are observed by a moving camera in the context of a continuously varying cluttered background. The green book tracked in the fourth sequence undergoes significant changes regarding its pose and shape, whereas light reflections on its glossy surface significantly affect its appearance. The fifth sequence is an example of a low quality video captured by a moving camera, illustrating the inherently deformable body of a caterpillar in motion. The sixth and seventh sequences show a human head and hand, respectively, which both go through extended pose variations in front of a complex background. Finally, the last, low resolution sequence has been captured by a medical endoscope. In this sequence, a target object is successfully tracked within a low-contrast background.

Each of the image sequences in Fig.(3) illustrating human hands or faces as well as the green book sequence consists of 400 frames of resolution 640×480 pixels, captured at a frame rate of 5-10 fps. The resolution of each frame of the image sequences illustrated in the second and the fifth row is 320×240 pixels. The last image sequence depicted in the Fig.(3), captured by a medical endoscope consists of 20 image frames of size 256×256 pixels each.

The reported experiments were generated based on a Matlab implementation, running on a PC equipped with an Intel i7 CPU and 4 GB of RAM memory. The runtime performance of the current implementation varies between 4 to 6 seconds per frame for 640×480 images. A near real-time runtime performance is feasible by optimizing both the EM-like component of the tracking method and the solution of the large sparse linear system of equations of the Random Walker formulation in the segmentation procedure.

Each frame shown in Fig.(3) is annotated with the results of the proposed algorithm and the results of the tracking method proposed in [14]. More specifically, the blue solid ellipse shows the expected position and coarse orientation of the tracked object as this results from the tracking part of the proposed methodology. The green solid object contour is the main result of the proposed algorithm which shows the fine object segmentation. Finally, the result of [14] is shown for comparison as a red dotted ellipse. Experimental results on the full video datasets are available online[1].

In all sequences, the appearance models of the tracked objects have been built based on the RGB color space. The object and background appearance models used to compute the prior color cues are color histograms with 32 bins per histogram for both the object and the background. Preserving the parameter configuration of the object tracking algorithm as described in [14], the target appearance model of the tracker is implemented by a color histogram of 8 bins per dimension.

The Random Walker segmentation method involves three different formulations to obtain the probabilities of each pixel to belong to each of the labels of the segmentation problem, as described in Sec. 2.2. The three options refer to the usage of seed pixels (pre-labeled graph nodes), prior values (probabilities/beliefs on label assignments for some graph nodes), or a combination of them. For the last option, the edge weights of the graph are computed by the Eq.(6), where the β scalar parameter controls the scale of the edgeness (color contrast) between adjacent graph nodes. The pixel-wise posterior values are computed using Bayesian inference as described in Sec. 2.2 and are exploited to guide segmentation as seed and prior values in terms of Random Walker terminology. Each pixel x_i of posterior value $P(L_l \mid x_i)$ greater or equal to 0.9 is considered as a seed pixel for the label L_l, thus as a seed node on the graph G. Any other pixel of posterior value $P(L_l \mid x_i)$ less than 0.9 is considered as a prior value for label L_l. In the case of prior values, the γ parameter is introduced to adjust the degree of authority of the prior beliefs towards the definite label-assignments expressed by

[1] `http://www.ics.forth.gr/~argyros/research/trackingsegmentation.htm`

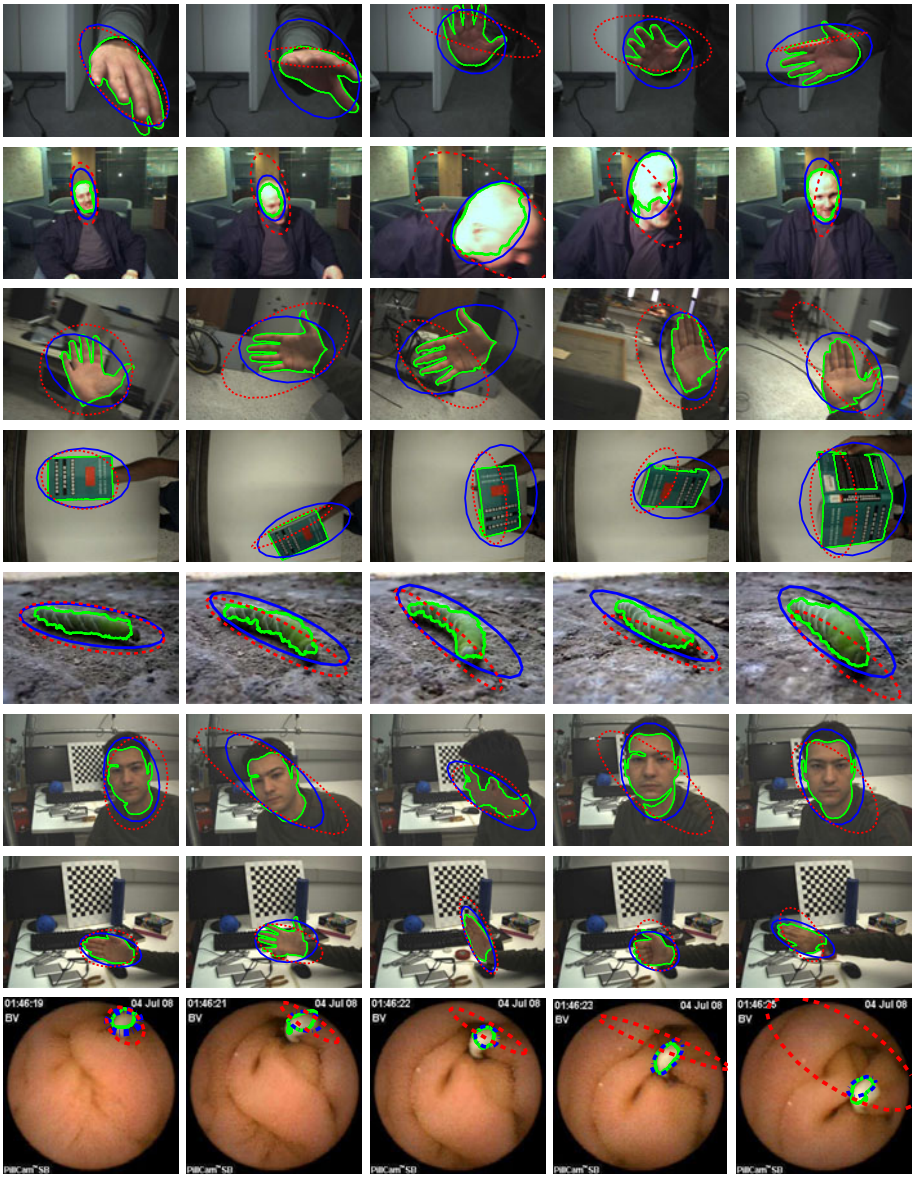


Fig. 3. Experimental results and qualitative comparison between the proposed framework providing tracking and segmentation results (blue solid ellipse and green solid object contour, respectively) and the tracking algorithm of [14] (red dotted ellipse). See text for details.

Table 1. Quantitative assessment of segmentation accuracy. See text for details.

Segmentation option	Precision	Recall	F-measure
Priors	93,5%	92,9%	93,1%
Seeds	97,5%	99,1%	98,3%
Priors and Seeds	97,5%	99,1%	98,3%

the seed nodes of the graph. In our experiments, the β parameter was selected within the interval of $[10 - 50]$, whereas the γ ranges within $[0.05 - 0.5]$.

In order to assess quantitatively the influence of the three different options regarding the operation of the Random Walker on the quality of segmentation results, the three different variants have been tested independently on an image sequence consisting of $1,000$ video frames. For each and every of these frames ground truth information is available in the form of a manually segmented foreground object mask. Table 1 summarizes the average (per frame) precision, recall and F-measure performance of the proposed algorithm compared to the ground truth. As it can be verified, although all three options perform satisfactorily, the use of seeds improves the segmentation performance.

4 Summary

In this paper we presented a method for online, joint tracking and segmentation of an non-rigid object in a monocular video, captured by a possibly moving camera. The proposed approach aspires to relax several limiting assumptions regarding the appearance and shape of the tracked object, the motion of the camera and the lighting conditions. The key contribution of the proposed framework is the efficient combination of an appearance-based tracking with Random Walker-based segmentation that jointly enables enhanced tracking performance and fine segmentation of the target object. A 2D affine transformation is computed to propagate the segmented object shape of the previous frame to the new frame exploiting the information provided by the ellipse region capturing the segmented object and the ellipse region predicted by the tracker in the new frame. A shape-band area is computed indicating an area of uncertainty where the true object boundaries lie in the new frame. Static image cues including pixel-wise color and spatial likelihoods are fused using Bayesian inference to guide the Random Walker-based object segmentation in conjunction with the color-contrast (edgeness) likelihoods between neighboring pixels. The performance of the proposed method is demonstrated in a series of challenging videos and in comparison with the results of the tracking method presented in [14].

Acknowledgments

This work was partially supported by the IST-FP7-IP-215821 project GRASP. The contributions of FORTH-ICS members I. Oikonomidis and N. Kyriazis to the development of the proposed method are gratefully acknowledged.

References

1. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. ACM Comput. Surv. 38, 13 (2006)
2. Isard, M., Blake, A.: Condensation: Conditional density propagation for visual tracking. International Journal of Computer Vision 29, 5–28 (1998)
3. Paragios, N., Deriche, R.: Geodesic active contours and level sets for the detection and tracking of moving objects. IEEE Transactions on PAMI 22, 266–280 (2000)
4. Yilmaz, A., Li, X., Shah, M.: Contour-based object tracking with occlusion handling in video acquired using mobile cameras. IEEE Transactions on PAMI 26, 1531–1536 (2004)
5. Bibby, C., Reid, I.: Robust real-time visual tracking using pixel-wise posteriors. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 831–844. Springer, Heidelberg (2008)
6. Khan, S., Shah, M.: Object based segmentation of video using color, motion and spatial information. In: IEEE Computer Society Conference on CVPR, vol. 2, p. 746 (2001)
7. Baltzakis, H., Argyros, A.A.: Propagation of pixel hypotheses for multiple objects tracking. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Kuno, Y., Wang, J., Pajarola, R., Lindstrom, P., Hinkenjann, A., Encarnação, M.L., Silva, C.T., Coming, D. (eds.) ISVC 2009. LNCS, vol. 5876, pp. 140–149. Springer, Heidelberg (2009)
8. Yu, T., Zhang, C., Cohen, M., Rui, Y., Wu, Y.: Monocular video foreground/background segmentation by tracking spatial-color gaussian mixture models. In: IEEE Workshop on Motion and Video Computing (2007)
9. Yin, Z., Collins, R.T.: Shape constrained figure-ground segmentation and tracking. In: IEEE Computer Society Conference on CVPR, pp. 731–738 (2009)
10. Ren, X., Malik, J.: Tracking as repeated figure/ground segmentation. In: IEEE Computer Society Conference on CVPR, pp. 1–8 (2007)
11. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE Transactions on PAMI 25, 564–577 (2003)
12. Tao, H., Sawhney, H., Kumar, R.: Object tracking with bayesian estimation of dynamic layer representations. IEEE Transactions on PAMI 24, 75–89 (2002)
13. Jepson, A.D., Fleet, D.J., El-Maraghi, T.F.: Robust online appearance models for visual tracking. IEEE Transactions on PAMI 25, 1296–1311 (2003)
14. Zivkovic, Z., Krose, B.: An em-like algorithm for color-histogram-based object tracking. In: IEEE Computer Society Conference on CVPR, vol. 1, pp. 798–803 (2004)
15. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: IEEE Computer Society Conference on CVPR, vol. 2, p. 2142 (2000)
16. Grady, L.: Multilabel random walker image segmentation using prior models. In: Proceedings of the 2005 IEEE Computer Society Conference on CVPR, vol. 1, pp. 763–770 (2005)
17. Grady, L.: Random walks for image segmentation. IEEE Transactions on PAMI 28, 1768–1783 (2006)
18. Grady, L., Funka-Lea, G.: Multi-label image segmentation for medical applications based on graph-theoretic electrical potentials. In: Sonka, M., Kakadiaris, I.A., Kybic, J. (eds.) CVAMIA/MMBIA 2004. LNCS, vol. 3117, pp. 230–245. Springer, Heidelberg (2004)
19. von Luxburg, U.: A tutorial on spectral clustering. Statistics and Computing 17, 395–416 (2007)
20. Boykov, Y., Funka-Lea, G.: Graph cuts and efficient n-d image segmentation. International Journal of Computer Vision 70, 109–131 (2006)